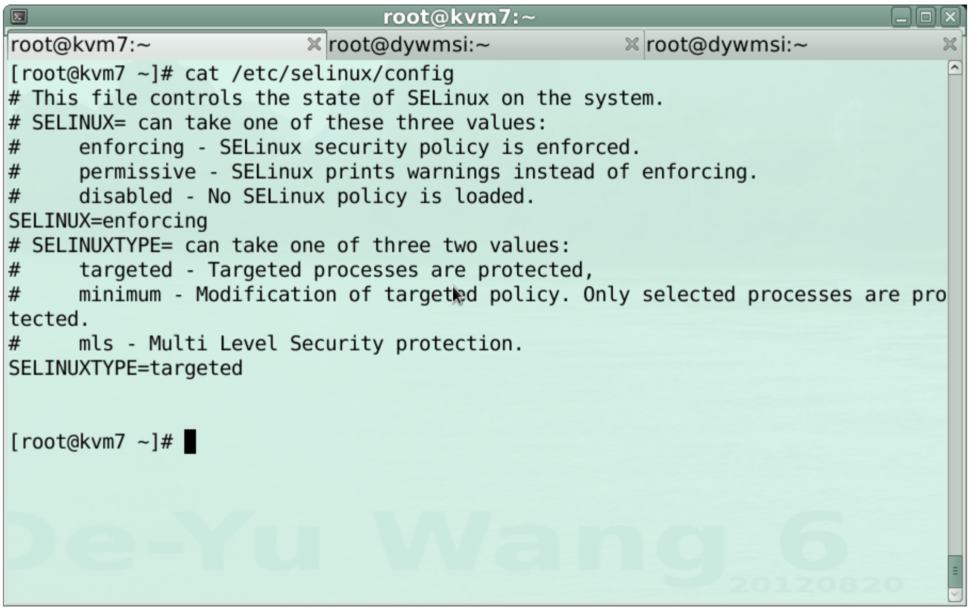
Linux 安全管理

Department of Computer Science and Information Engineering
Chaoyang University of Technology
Taichung, Taiwan, Republic of China

Instructor: De-Yu Wang (王德譽)
E-mail: dywang@csie.cyut.edu.tw
Homepage: http://dywang.csie.cyut.edu.tw
Phone: (04)23323000 ext 4538
Office: E738



May 14, 2020

- Instructor: De-Yu Wang
    1. Email: dywang@csie.cyut.edu.tw
    2. Homepage: https://dywang.csie.cyut.edu.tw
    3. 本文件 pdf 檔下載
    4. Phone: (04)23323000 ext 4538
    5. Office: E738
- 參考資料
    1. SELinux User's and Administrator's Guide。

Contents

Chapter 1

SELinux

1.1 前言

1. 自主式存取控制 (Discretionary Access Control, DAC)
   (a) 傳統的檔案權限與帳號關係：依據程序的擁有者與檔案資源的 rwx 權限來決定有無存取的能力。
   (b) 若該程序屬 root 權限，則其可在系統上進任何資源存取。
   (c) 若某個目錄權限爲 777，則任何帳號的程序都可任意存取及寫入。
2. 系統出現問題
   (a) 內部員工的資源誤用遠高於外部攻擊。
   (b) 員工資源誤用：系統管理員爲了自己方便，將防火牆完全關閉或某個檔案目錄的權限設定爲 777。
   (c) 爲控管員工資源誤用問題，美國國家安全局 (NSA) 開發安全增強式 Linux（SELinux, Security-Enhanced Linux）模組，並整合到 Linux 核心。
   (d) SELinux 是一種強制存取控制（mandatory access control, MAC）的實現。
3. 強制存取控制
   (a) 針對特定的程序與特定的檔案資源進行權限的控管。
   (b) WWW server 啓動的程序爲 httpd，selinux 預設僅能在/var/www 目錄下存取檔案。
4. 開啓 SELinux 出現的現象：
   (a) httpd 配置正常，防火牆也放行，但用户就是無法訪問？
   (b) vsftpd 配置正常，防火牆也放行，但客户端訪問提示無法轉換家目錄？
   (c) 關閉 SELinux 後一切正常，因此很多人對 SELinux 有了極大的偏見，認爲 SELinux 只有帶來麻煩，於是選擇在安裝系統後或出現問題時關閉 SELinux。
   (d) 實際上，SELinux 在很多時候提供關鍵的保障，所以還是要開啓 SELinux，出現無法訪問等問題時，以修改 SELinux 安全上下文、布林值... 等方式來解決。

1.2 SELinux Contexts

1. Linux 執行程序或檔案會以 SELinux 上下文 (Contexts) 做標記，內容包含：SELinux user, role, type, level，以這些資訊來控制存取決定。

2. ls -Z 查詢檔案的 SELinux contexts，列出四個欄位 user:role:type:level，分別以冒號隔開。

```
[root@kvm7 ~]# ls -Z anaconda-ks.cfg
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
```

   (a) user：使用者身份，以確認 SELinux 策略，Linux user 帳號會被對應到 SELinux user，執行 semanage login -l 可以列出兩者的對應，如果無任何回應，表示/etc/selinux/targeted/active/seusers 這個檔案不存在，seusers 是由 selinux-policy-targeted 套件提供，重新安裝套件即可。

```
[root@kvm7 ~]# yum reinstall selinux-policy-targeted
```

   其中 __default__ 表示登入的用户，一般是使用 useradd 增加的帳號；system_u 表示系統程序，由執行使用者可以確認身份。

```
[root@kvm7 ~]# semanage login -l
Login Name           SELinux User         MLS/MCS Range
    Service
__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

   (b) role：SELinux 以 Role-Based Access Control (RBAC) 控制存取，角色 (role) 決定是否可存取 domain，而每個 user 可以有多個角色，但同一時間只能有一個角色。

   (c) type：指定資料型別，規則中定義何種程序型別訪問何種檔案。

   (d) level：Multi-Level Security (MLS) and Multi-Category Security (MCS)，僅在 MLS 策略下才有用。

3. SELinux Contexts 四個欄位說明：

項目	用户 (user)	角色 (role)	類型 (
程序	system_u 爲系統服務程序，受到管制。unconfined_u 爲不管制程序，通常都是用户自己開啓的，如 bash。	system_r 爲系統服務程序，受到管制。unconfined_r 爲不管制程序，通常都是用户自己開啓的，如 bash。	在 tar 下， 類型 代表 才能
檔案	system_u 爲系統服務創建的檔案。unconfined_u 爲用户自己創建的檔案。	均爲 object_r，表示一個檔案。	在 tar 下， 類型 (不代 程序 案。

1.3 啓動、關閉與觀察

1. SELinux 開機啓動模式

```
[root@kvm7 ~]# vim /etc/selinux/config
[root@kvm7 ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
    are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2. 關察目前 SELinux 狀態

```
[root@kvm7 ~]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

3. 改變 SELinux 模式

```
[root@kvm7 ~]# getenforce
Permissive
[root@kvm7 ~]# setenforce 1
[root@kvm7 ~]# getenforce
Enforcing
```

1.4 SELinux 執行策略

1. 配置檔案/etc/selinux/config 還包含了 SELinux 執行策略的資訊，透過改變變數 SELINUXTYPE 的值實現：

```
[root@kvm7 ~]# tail -5 /etc/selinux/config
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes
    are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2. 策略說明
   (a) targeted：預設使用的策略，對大部分網絡服務程序進行管制。
   (b) minimum：以 targeted 爲基礎，僅對選定的網絡服務程序進行管制。
   (c) mls：多級安全保護，對所有的程序進行管制，是最嚴格的策略，配置難度非常大。一般不用，除非對安全性有極高的要求。

3. 查詢系統只安裝 targeted 策略。

```
[root@kvm7 ~]# ll -lrt /etc/selinux/
total 12
-rw-r--r--. 1 root root 2321 Jun 10  2014 semanage.conf
drwxr-xr-x. 6 root root 4096 Dec 28  2018 targeted
-rw-r--r--. 1 root root  546 Oct 14 19:24 config
```

4. 若要使用 mls 策略必須安裝安裝套件。

```
[root@kvm7 ~]# yum install selinux-policy-mls
```

1.5 SELinux 基本用法

1. coreutils 的工具如 ps、ls 可以增加 Z 選項的方式取得 SELinux 的資訊。

```
[root@kvm7 ~]# ls -Z anaconda-ks.cfg
-rw-------. root root system_u:object_r:admin_home_t:s0 anaconda-ks.cfg
[root@kvm7 ~]# ps auxZ | grep apache
system_u:system_r:httpd_t:s0  apache  15694  0.0  0.2 215320  3040 ?  S
    Oct14   0:00 /usr/sbin/httpd -DFOREGROUND
```

2. chcon 修改檔案或目錄的 SELinux contexts。

```
[root@kvm7 ~]# chcon --help
Usage: chcon [OPTION]... CONTEXT FILE...
  or:  chcon [OPTION]... [-u USER] [-r ROLE] [-l RANGE] [-t TYPE] FILE
      ...
  -u, --user=USER        set user USER in the target security context
  -r, --role=ROLE        set role ROLE in the target security context
  -t, --type=TYPE        set type TYPE in the target security context
  -l, --range=RANGE      set range RANGE in the target security context
  -R, --recursive        operate on files and directories recursively
  -v, --verbose          output a diagnostic for every file processed
```

3. chcon 範例

```
[root@kvm7 ~]# ls -Z /etc/resolv.conf
-rw-r--r--. root root system_u:object_r:net_conf_t:s0  /etc/resolv.conf
[root@kvm7 ~]# cp /etc/resolv.conf .
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 resolv.conf
[root@kvm7 ~]# chcon -t net_conf_t resolv.conf
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:net_conf_t:s0 resolv.conf
```

4. restorecon 恢復檔案或目錄的 SELinux contexts 到預設值。

```
[root@kvm7 ~]# restorecon
usage:  restorecon [-iFnprRv0] [-e excludedir] pathname...
usage:  restorecon [-iFnprRv0] [-e excludedir] -f filename
```

5. restorecon 範例

```
[root@kvm7 ~]# restorecon resolv.conf
[root@kvm7 ~]# ls -Z resolv.conf
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 resolv.conf
```

6. getsebool 查詢 SELinux 布林值型規則的狀態。

```
[root@kvm7 ~]# getsebool
usage:  getsebool -a or getsebool boolean...
```

7. getsebool 範例

```
[root@kvm7 ~]# getsebool -a | grep ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
sftpd_write_ssh_home --> off
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

8. setsebool 開關 SELinux 布林值型規則。

```
[root@kvm7 ~]# setsebool
Usage:  setsebool [ -NPV ] boolean value | bool1=val1 bool2=val2...
```

9. setsebool 範例，沒加 -P 選項只是現在生效，重開機後會恢復到原設定值。

```
[root@kvm7 ~]# setsebool sftpd_write_ssh_home on
[root@kvm7 ~]# getsebool -a | grep ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
sftpd_write_ssh_home --> on
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

10. semanage 設定 SELinux 策略的某些元素，例如：檔案 context、程序埠號等。

```
[root@kvm7 ~]# semanage
usage: semanage [-h]

                {...,port,..,fcontext,...}
                ...
semanage is used to configure certain elements of SELinux policy with-
    out
requiring modification to or recompilation from policy source.
```

```

    port              Manage network port type definitions
    fcontext          Manage file context mapping definitions
    boolean           Manage booleans to selectively enable functionality
```

11. semanage 範例一：添加目錄的 SELinux context。

```
[root@kvm7 ~]# mkdir /usr/share/www
[root@kvm7 ~]# ls -Z /usr/share/www
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   /usr/share/www
[root@kvm7 ~]# semanage fcontext -a -t httpd_sys_content_t "/usr/share/www(/.*)?"
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   /usr/share/www
[root@kvm7 ~]# restorecon -Rv /usr/share/www
restorecon reset /usr/share/www context unconfined_u:object_r:usr_t:s0
->unconfined_u:object_r:httpd_sys_content_t:s0
[root@kvm7 ~]# ls -Zd /usr/share/www
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /usr/share/www
```

12. semanage 範例二：添加 httpd 程序允許訪問的 port。

```
[root@kvm7 ~]# semanage port -l | grep ^http_port
http_port_t      tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
[root@kvm7 ~]# semanage port -a -t http_port_t -p tcp 8989
[root@kvm7 ~]# semanage port -l | grep ^http_port
http_port_t      tcp      8989, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

13. semanage 範例三：刪除 httpd 程序允許訪問的 port。

```
[root@kvm7 html]# semanage port -d -t http_port_t -p tcp 8989
[root@kvm7 html]# semanage port -l | grep ^http_port
http_port_t       tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

1.6 SEtools

1. seinfo 可查詢 SELinux Context type，先查 seinfo 命令由哪個套件提供？

```
[root@kvm5 ~]# yum provides *bin/seinfo
....................
Matched from:
```

```
Filename      : /usr/bin/seinfo
```

2. seinfo 命令由 setools-console 提供，安裝此套件。

```
[root@kvm5 ~]# yum install setools-console
```

3. 執行 seinfo -t 列出所有 selinux context type，再以管線處理及 grep 過濾 samba 相關的 type。

```
[root@kvm5 ~]# seinfo -t | grep samba
   samba_secrets_t
   samba_spool_t
   samba_unconfined_script_exec_t
   samba_net_t
   samba_var_t
   samba_net_exec_t
   samba_net_tmp_t
   samba_unconfined_net_t
   samba_unconfined_script_t
   samba_unit_file_t
   sambagui_exec_t
   samba_share_t
   samba_initrc_exec_t
   sambagui_t
   samba_etc_t
   samba_log_t
```

Chapter 2

SELinux 管理 Confined 服務

2.1 前言

1. 一般而言，登入用戶運作在不受限 (Unconfined) `unconfined_t` domain，由 init 啓動的系統程序運作於不受限 `unconfined_service_t` domain。
2. 幾乎每種服務都接受網路上的請求，例如：sshd, httpd, vsftpd 等都是受限的 (Confined)。
3. 受限服務在自己的 domain 運作，像 httpd 服務運作在 `httpd_t` domain。
4. 可執行和可寫入記憶體同時適用於於受限域和未受限 domains。
5. 預設狀況，運作於不受限 domain 可配置記憶體並執行；受限 domain 則透過 SELinux boolean 值的開關檢查並決定是否可配置記憶體及執行。

2.2 Httpd SELinux Context

1. 安裝套件。

```
[root@kvm7 ~]# yum install httpd -y
```

2. 啓動 httpd 服務，並設定開機啓動。

```
[root@kvm7 ~]# systemctl enable httpd.service
[root@kvm7 ~]# systemctl start httpd
```

3. 查看 httpd 服務的 context 的 type 是 `httpd_t`，運作在 `httpd_t` domain。

```
[root@kvm5 ~]# ps -eZ | grep httpd
system_u:system_r:httpd_t:s0    10803 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0    10804 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0    10805 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0    10806 ?        00:00:00 httpd
```

```
system_u:system_r:httpd_t:s0      10807 ?        00:00:00 httpd
system_u:system_r:httpd_t:s0      10808 ?        00:00:00 httpd
```

4. 切換工作目錄到網頁的根目錄/var/www/html。

```
[root@kvm7 ~]# cd /var/www/html/
```

5. cal 印出月曆，分別存於 root 家目錄及/var/www/html 目錄。

```
[root@kvm7 html]# cal > ~/index.html
[root@kvm7 html]# cal > index1.html
```

6. 將 root 家目錄上的 index.html 移到/var/www/html 目錄。

```
[root@kvm7 html]# mv ~/index.html .
```

7. 查看/var/www/html 目錄上的 index.html 及 index1.html 的 SELinux contexts，index.html 是由 root 家目錄移過去的，所以是 `admin_home_t`。

```
[root@kvm7 html]# ls -Z index*.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index1.html
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 index.html
```

8. index1.html 的 SELinux context 是 `httpd_sys_content_t`，所以網頁訪問成功，列出月曆。

```
[root@kvm7 html]# curl http://127.0.0.1/index1.html
    October 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

9. index.html 的 SELinux context 是 `admin_home_t`，所以出現沒有權限存取的訊息，網頁無法訪問。

```
[root@kvm7 html]# curl http://127.0.0.1/index.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /index.html
on this server.</p>
</body></html>
```

10. restorecon 將/var/www/html/ 目錄下的 index.html 恢復到預設的 SELinux context `httpd_sys_content_t`。

```
[root@kvm7 html]# restorecon index.html
[root@kvm7 html]# ls -Z index*.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
    index1.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index
    .html
```

11. 再次連線網頁 index.html，已可成功列出月曆。

```
[root@kvm7 html]# curl http://127.0.0.1/index.html
    October 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

2.3 Httpd SELinux Port

1. 查詢 `http_port_t`，8989 port 不是 httpd 程序允許訪問的 port。

```
[root@kvm7 html]# semanage port -l | grep -w http_port
http_port_t        tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

2. 編輯 httpd 服務配置檔，將服務的訪問埠號改成 8989。

```
[root@kvm7 html]# vim /etc/httpd/conf/httpd.conf
[root@kvm7 html]# grep ^Listen /etc/httpd/conf/httpd.conf
Listen 8989
```

3. 無法重新啓動 httpd 服務。

```
[root@kvm7 html]# systemctl restart httpd.service
Job for httpd.service failed. See 'systemctl status httpd.service' and 'journalctl -xn' for details.
```

4. 查看 httpd 服務狀態。

```
[root@kvm7 html]# systemctl status httpd.service
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: failed (Result: exit-code) since Thu 2019-10-17 07:23:04 CST; 15s ago
....
Oct 17 07:23:04 kvm7.deyu.wang systemd[1]: Starting The Apache HTTP Server...
Oct 17 07:23:04 kvm7.deyu.wang httpd[17917]: (13)Permission denied: AH00072:...9
Oct 17 07:23:04 kvm7.deyu.wang httpd[17917]: (13)Permission denied: AH00072:...9
```

5. 新增 8989 爲 httpd 服務允許訪問的埠號。

```
[root@kvm7 html]# semanage port -a -t http_port_t -p tcp 8989
```

6. 再重新啓動 httpd 服務，成功啓動服務。

```
[root@kvm7 html]# systemctl restart httpd.service
```

7. 測試以 8989 埠號連線網頁，成功列出月曆。

```
[root@kvm7 html]# curl http://127.0.0.1:8989/index.html
    October 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

2.4 NFS Server

1. NFS 與 SELinux
   (a) 預設情況 NFS 是受限服務，但 nfsd 服務被標記爲不受限的 `kernel_t` doamin 型態。
   (b) SELinux 策略預設允許 NFS 分享檔案，而且支援 server 傳遞 SELinux 標記到 client，以提供對 NFS 更安全的控制。

2. 安裝 nfs-utils 套件。

```
[root@kvm5 ~]# yum install -y nfs-utils
```

3. 建立 NFS 分享目錄/public。

```
[root@kvm5 ~]# mkdir /public
```

4. 查看分享目錄/public 的 SELinux Context 爲 `default_t`。

```
[root@kvm5 ~]# ls -Zd /public
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /public
```

5. 修改 NFS 分享目錄 (含次目錄) 的 SELinux context 爲 `public_content_t`。

```
[root@kvm5 ~]# semanage fcontext -a -t public_content_t "/public(/.*)?"
```

6. 將/public (含次目錄) 的 SELinux context 恢復爲預設值，也就是剛剛設定的 `public_content_t`。

```
[root@kvm5 ~]# restorecon -Rv /public
restorecon reset /public context unconfined_u:object_r:default_t:s0->
    unconfined_u:object_r:public_content_t:s0
```

7. 編輯 nfs 服務配置檔，將/public 目錄分享給 192.168.122.0/24 網段。

```
[root@kvm5 ~]# vim /etc/exports
[root@kvm5 ~]# cat /etc/exports
/public 192.168.122.0/24(ro,sync)
```

8. 啓動 nfs 服務，並設定開機啓動。

```
[root@kvm5 ~]# systemctl enable nfs-server.service
[root@kvm5 ~]# systemctl start nfs-server.service
```

9. 查詢 nfs 服務支援的版本，必須有 4.2 版以支援傳遞 SELinux Contexts 到 client。

```
[root@kvm5 ~]# cat /proc/fs/nfsd/versions
-2 +3 +4 +4.1 +4.2
```

10. 輸出所有設定的 nfs 分享目錄。

```
[root@kvm5 ~]# exportfs -arv
exporting 192.168.122.0/24:/public
```

2.5 NFS 伺服器防火牆設定

1. 永久增加 nfs 服務到 firewall。

```
[root@kvm5 ~]# firewall-cmd --permanent --add-service=nfs
success
```

2. 永久增加 rpc-bind 服務到 firewall。

```
[root@kvm5 ~]# firewall-cmd --permanent --add-service=rpc-bind
success
```

3. 永久增加 mountd 服務到 firewall。

```
[root@kvm5 ~]# firewall-cmd --permanent --add-service=mountd
success
```

4. 重新載入防火牆設定。

```
[root@kvm5 ~]# firewall-cmd --reload
success
```

5. 列出 firewall 已新增 nfs, rpc-bind, mountd 等服務。

```
[root@kvm5 ~]# firewall-cmd --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client mountd nfs rpc-bind ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

2.6 NFS Client 端掛載

1. Client 端列出 server 端 kvm5.deyu.wang 分享的 NFS 目錄。

```
[root@kvm7 ~]# showmount -e kvm5.deyu.wang
Export list for kvm5.deyu.wang:
/public    192.168.122.0/24
```

2. 建立掛載目錄

```
[root@kvm7 ~]# mkdir /mnt/nfsmount
```

3. 以 4.2 版本手動掛載 NFS 分享目錄

```
[root@kvm7 ~]# mount.nfs4 -o v4.2 kvm5.deyu.wang:/public /mnt/nfsmount
```

4. 查看掛載目錄的 SELinux Context 的 type 為 `public_content_t`，成功由 server 傳遞到 client。

```
[root@kvm7 ~]# ls -Zd /mnt/nfsmount/
drwxr-xr-x. root root unconfined_u:object_r:public_content_t:s0 /mnt/
    nfsmount/
```

5. 改以 4.0 版本手動掛載 NFS 分享目錄

```
[root@kvm7 ~]# mount -o v4.0 kvm5.deyu.wang:/public /mnt/nfsmount
```

6. 查看掛載目錄的 SELinux Context 的 type 爲 `nfs_t`，無法由 server 傳遞到 client。

```
[root@kvm7 ~]# ls -Zd /mnt/nfsmount/
drwxr-xr-x. root root system_u:object_r:nfs_t:s0       /mnt/nfsmount/
```

2.7 Httpd 使用 NFS

1. 編輯 httpd 配置檔，將 DocumentRoot 改到/mnt/nfsmount。

```
[root@kvm7 ~]# vim /etc/httpd/conf/httpd.conf
[root@kvm7 ~]# grep -A5 ^DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot "/mnt/nfsmount"
<Directory "/mnt/nfsmount">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

2. 重新啓動 httpd 服務。

```
[root@kvm7 ~]# systemctl restart httpd.service
```

3. 在 NFS server 的/public 目錄下新增 index.html。

```
[root@kvm5 ~]# cal > /public/index.html
```

4. 查看/mnt/nfsmount/index.html 的 SELinux contexts 爲 `nfs_t`。

```
[root@kvm7 ~]# ls -Z /mnt/nfsmount/
-rw-r--r--. root root system_u:object_r:nfs_t:s0       index.html
```

5. curl 訪問網頁，回傳沒有權限存取。

```
[root@kvm7 ~]# curl http://127.0.0.1/index.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /index.html
on this server.</p>
```

```
</body></html>
```

6. index.html 的 SELinux context 是 `nfs_t`，必須開啓 `httpd_use_nfs` 布林值，查看此布林值目前是 off。如果是 `cifs_t` 則要開啓 `httpd_use_cifs` 布林值。

```
[root@kvm7 ~]# getsebool -a | grep httpd_use_nfs
httpd_use_nfs --> off
```

7. 開啓布林值 `httpd_use_nfs`。

```
[root@kvm7 ~]# setsebool -P httpd_use_nfs on
[root@kvm7 ~]# getsebool -a | grep httpd_use_nfs
httpd_use_nfs --> on
```

8. curl 再訪問網頁，成功回傳網頁內容。

```
[root@kvm7 ~]# curl http://127.0.0.1/index.html
    October 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

2.8 SAMBA 架設

1. 安裝 samba 套件

```
[root@kvm5 ~]# yum -y install samba
```

2. 在 kvm5.deyu.wang 主機再新增一共享目錄/groupdir。

```
[root@kvm5 ~]# mkdir /groupdir
```

3. 在共享目錄/groupdir 下產生兩個空檔案。

```
[root@kvm5 ~]# touch /groupdir/file1
[root@kvm5 ~]# touch /groupdir/file2
```

4. 設定目錄/groupdir 下的 selinux context type 爲 `samba_share_t`，不先查詢，使用 TAB 鍵一樣會出現提示及補齊。

```
[root@kvm5 ~]# semanage fcontext -a -t samba_share_t '/groupdir(/.*)?'
```

5. 恢復目錄/groupdir 下的 selinux type 爲 `samba_share_t`。

```
[root@kvm5 ~]# restorecon -Rv /groupdir
restorecon reset /groupdir context unconfined_u:object_r:default_t:s0->
    system_u:object_r:samba_share_t:s0
```

6. 編輯配置檔/etc/samba/smb.conf，建立分享目錄/groupdir 的名稱爲 common、可瀏覽、只允許 192.168.122.0/24 網域用户存取、用户 deyu1 可以使用 123qwe 密碼存取這個分享目錄。

```
[root@kvm5 ~]# vim /etc/samba/smb.conf
[root@kvm5 ~]# tail -5 /etc/samba/smb.conf
[common]
    path = /groupdir
    browseable = yes
    hosts allow = 192.168.122.
```

7. 測試配置檔/etc/samba/smb.conf 看是否有錯誤？

```
[root@kvm5 ~]# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit
    (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[common]"
Loaded services file OK.
Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters
[global]
    printcap name = cups
    security = USER
```

```
    workgroup = SAMBA
    idmap config * : backend = tdb
    cups options = raw

[homes]
    browseable = No
    comment = Home Directories
    inherit acls = Yes
    read only = No
    valid users = %S %D%w%S

[printers]
    browseable = No
    comment = All Printers
    create mask = 0600
    path = /var/tmp
    printable = Yes

[print$]
    comment = Printer Drivers
    create mask = 0664
    directory mask = 0775
    force group = @printadmin
    path = /var/lib/samba/drivers
    write list = @printadmin root


[common]
    hosts allow = 192.168.122.
    path = /groupdir
```

8. 要建立 smb 用户必須先安裝 samba-client 套件。

```
[root@kvm5 ~]# yum -y install samba-client
```

9. 如果 deyu1 用户不存在，則新增一個用户 deyu1。

```
[root@kvm5 ~]# useradd deyu1
```

10. 新增 deyu1 爲 samba 用户。

```
[root@kvm5 ~]# smbpasswd -a deyu1
New SMB password:
Retype new SMB password:
Added user deyu1.
```

11. 查看 samba 用户。

```
[root@kvm5 ~]# pdbedit -L
deyu1:1000:
```

12. 設定開機啓動 smb 服務。

```
[root@kvm5 ~]# systemctl enable smb.service
ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-
    user.target.wants/smb.service'
```

13. 啓動 smb 服務。

```
[root@kvm5 ~]# systemctl start smb.service
```

14. smbclient 以 deyu1 登入 kvm5.deyu.wang 分享的 common。

```
[root@kvm5 ~]# smbclient -L //kvm5.deyu.wang/common -U deyu1%123qwe

    Sharename       Type      Comment
    ---------       ----      -------
    print$          Disk      Printer Drivers
    common          Disk
    IPC$            IPC       IPC Service (Samba 4.7.1)
    deyu1           Disk      Home Directories
Reconnecting with SMB1 for workgroup listing.

    Server               Comment
    ---------            -------

    Workgroup            Master
    ---------            -------
    SAMBA                KVM5
```

2.9 防火牆設定

1. 防火牆永久加入 samba 服務。

```
[root@kvm5 ~]# firewall-cmd --permanent --add-service=samba
success
```

2. 重新載入防火牆。

```
[root@kvm5 ~]# firewall-cmd --reload
success
```

3. 列出防火牆設定，支援的服務包含 samba。

```
[root@kvm5 ~]# firewall-cmd --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client samba ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

2.10 測試 SMB 分享目錄

1. 在 kvm7.deyu.wang 主機測試，因要用到 mount.cifs 命令，所以必須先安裝 cifs-utils 套件。

```
[root@kvm7 ~]# yum -y install cifs-utils samba-client
```

2. 先測試 samba 用户 deyu1 是否可以存取 kvm5.deyu.wang smb 分享目錄/common。

```
[root@kvm7 ~]# smbclient -L //kvm5.deyu.wang/common -U deyu1
Enter SAMBA\deyu1's password:

    Sharename       Type      Comment
    ---------       ----      -------
    print$          Disk      Printer Drivers
    common          Disk
    IPC$            IPC       IPC Service (Samba 4.7.1)
    deyu1           Disk      Home Directories
Reconnecting with SMB1 for workgroup listing.

    Server               Comment
    ---------            -------

    Workgroup            Master
    ---------            -------
    SAMBA                KVM5
```

3. 建立用户 deyu1 要掛載 smb 分享目錄的掛載目錄/mnt/deyu1。

```
[root@kvm7 ~]# mkdir /mnt/deyu1
```

4. 用户 deyu1 掛載 smb server kvm5.deyu.wang 分享目錄/common。

```
[root@kvm7 ~]# mount -o username=deyu1 //kvm5.deyu.wang/common /mnt/
    deyu1
Password for deyu1@//kvm5.deyu.wang/common:  ******
```

5. 成功掛載到/mnt/deyu1。

```
[root@kvm7 ~]# df -h /mnt/deyu1
Filesystem               Size  Used Avail Use% Mounted on
//kvm5.deyu.wang/common  3.1G  1.5G  1.4G  53% /mnt/deyu1
```

6. 查看掛載的/mnt/deyu1 目錄。

```
[root@kvm7 ~]# ls -Z /mnt/deyu1/
-rw-r--r--. root root system_u:object_r:cifs_t:s0      file1
-rw-r--r--. root root system_u:object_r:cifs_t:s0      file2
```

7. 如果 samba server 分享目錄的 SELinux Context 沒有設定成 `samba_share_t`，則回應拒絕存取。

```
[root@kvm7 ~]# ls -Z /mnt/deyu1/
ls: reading directory /mnt/deyu1/: Permission denied
```

2.11 Httpd 使用 SAMBA

1. 編輯 httpd 配置檔，將 DocumentRoot 改到/mnt/deyu1。

```
[root@kvm7 ~]# vim /etc/httpd/conf/httpd.conf
[root@kvm7 ~]# grep -A5 ^DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot "/mnt/deyu1"
<Directory "/mnt/deyu1">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

2. 重新啓動 httpd 服務。

```
[root@kvm7 ~]# systemctl restart httpd.service
```

3. 在 SAMBA server 的/groupdir 目錄下新增 index.html。

```
[root@kvm5 ~]# cal > /groupdir/index.html
```

4. 查看/mnt/nfsmount/index.html 的 SELinux contexts 爲 `cifs_t`。

```
[root@kvm7 ~]# ls -Z /mnt/deyu1/
-rw-r--r--. root root system_u:object_r:cifs_t:s0      file1
-rw-r--r--. root root system_u:object_r:cifs_t:s0      file2
-rw-r--r--. root root system_u:object_r:cifs_t:s0      index.html
```

5. curl 訪問網頁，回傳沒有權限存取。

```
[root@kvm7 ~]# curl http://127.0.0.1/index.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /index.html
on this server.</p>
</body></html>
```

6. index.html 的 SELinux context 是 `cifs_t`，必須開啓 `httpd_use_cifs` 布林值，查看此布林值目前是 off。

```
[root@kvm7 ~]# getsebool -a | grep httpd_use_cifs
httpd_use_cifs --> off
```

7. 開啓 `httpd_use_cifs` 布林值。

```
[root@kvm7 ~]# setsebool -P httpd_use_cifs on
[root@kvm7 ~]# getsebool -a | grep httpd_use_cifs
httpd_use_cifs --> on
```

8. curl 再訪問網頁，成功回傳網頁內容。

```
[root@kvm7 ~]# curl http://127.0.0.1/index.html
    October 2019
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Chapter 3

CHROOT

3.1 CHROOT 功用

1. 在 Linux 系統中目錄結構都是以根 (root) / 開始，使用 chroot (change root directory) 命令將系統的根目錄改變到指定的位置。
2. 變更根目錄後，無法存取新根目錄以外的目錄及檔案，大大提高安全性。
3. 系統維修時，用光碟或隨身碟開機，再改變根目錄至要維修的系統，進行維修。

3.2 建立 chroot 系統

1. chroot 指令來自 coreutils 套件，系統預設已安裝。
2. 建立要 chroot 的目錄。

```
[root@kvm5 ~]# mkdir /var/www/script
```

3. 切換目錄到/var/www/script

```
[root@kvm5 ~]# cd /var/www/script
```

4. 先查看 Linux 系統根目錄下幾個主要目錄的 SELinux Contexts。

```
[root@kvm5 script]# ls -Zd /{root,bin,dev,etc,home,lib64,proc,tmp,usr,var}
lrwxrwxrwx. root root system_u:object_r:bin_t:s0       /bin -> usr/bin
drwxr-xr-x. root root system_u:object_r:device_t:s0    /dev
drwxr-xr-x. root root system_u:object_r:etc_t:s0       /etc
drwxr-xr-x. root root system_u:object_r:home_root_t:s0 /home
lrwxrwxrwx. root root system_u:object_r:lib_t:s0       /lib64 -> usr/lib64
dr-xr-xr-x. root root system_u:object_r:proc_t:s0      /proc
dr-xr-x---. root root system_u:object_r:admin_home_t:s0 /root
```

```
drwxrwxrwt. root root system_u:object_r:tmp_t:s0       /tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0       /usr
drwxr-xr-x. root root system_u:object_r:var_t:s0       /var
```

5. 建立新 Linux 系統主要目錄。

```
[root@kvm5 script]# mkdir -p {root,dev,etc,home,proc,tmp,usr/{bin,lib64
    },var}
[root@kvm5 script]# ln -s usr/bin bin
[root@kvm5 script]# ln -s usr/lib64 lib64
```

6. 查看新建立的目錄 SELinux Contexts，都是 httpd_sys_context。

```
[root@kvm5 script]# ls -Z .
lrwxrwxrwx. root root unconfined_u:object_r:httpd_sys_content_t:s0 bin
    -> usr/bin
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 dev
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 etc
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 home
lrwxrwxrwx. root root unconfined_u:object_r:httpd_sys_content_t:s0 lib64
     -> usr/lib64
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 proc
dr-xr-x---. root root unconfined_u:object_r:httpd_sys_content_t:s0 root
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 tmp
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 usr
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 var
```

7. semanage 新增這些新建立目錄的 SELinux Contexts。

```
[root@kvm5 script]# for i in bin etc proc tmp usr var
> do
> semanage fcontext -a -t ${i}_t `pwd`/$i
> done
[root@kvm5 script]# semanage fcontext -a -t admin_home_t `pwd`/root
[root@kvm5 script]# semanage fcontext -a -t lib_t `pwd`/lib64
[root@kvm5 script]# semanage fcontext -a -t home_root_t `pwd`/home
[root@kvm5 script]# semanage fcontext -a -t device_t `pwd`/dev
```

8. restorecon 恢復這些新建立目錄的 SELinux Contexts。

```
[root@kvm5 script]# restorecon -Rv .
```

9. 查看新建立目錄的 SELinux Contexts，已符合系統要求。

```
[root@kvm5 script]# ls -Z .
lrwxrwxrwx. root root unconfined_u:object_r:bin_t:s0   bin -> usr/bin
drwxr-xr-x. root root unconfined_u:object_r:device_t:s0 dev
drwxr-xr-x. root root unconfined_u:object_r:etc_t:s0   etc
drwxr-xr-x. root root unconfined_u:object_r:home_root_t:s0 home
lrwxrwxrwx. root root unconfined_u:object_r:lib_t:s0   lib64 -> usr/
    lib64
drwxr-xr-x. root root unconfined_u:object_r:proc_t:s0  proc
drwxr-xr-x. root root unconfined_u:object_r:tmp_t:s0   tmp
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   usr
drwxr-xr-x. root root unconfined_u:object_r:var_t:s0   var
```

10. 複製 bin 下需要的執行檔，維持原檔案的 SELinux Context。

```
[root@kvm5 script]# cp --preserve=context \
/bin/{awk,bash,cat,echo,env,grep,mount,sed,sh} bin
[root@kvm5 script]# cp --preserve=context \
/usr/bin/{perl,php,python,tee,timeout,tr} usr/bin
```

11. 複製函式庫，一樣維持 SELinux Context。

```
[root@kvm5 script]# cp -a --preserve=context /usr/{lib64,lib,libexec}
    usr/
```

12. 先查看 null,zero,random,urandom 等 characters 的 major,minor numbers。

```
[root@kvm5 script]# stat -c 'major: %t minor: %T' \
/dev/{null,zero,random,urandom}
major: 1 minor: 3
major: 1 minor: 5
major: 1 minor: 8
major: 1 minor: 9
```

13. 建立 character device node。

```
[root@kvm5 script]# mknod -m 666 dev/null c 1 3
[root@kvm5 script]# mknod -m 666 dev/zero c 1 5
[root@kvm5 script]# mknod -m 666 dev/random c 1 8
[root@kvm5 script]# mknod -m 666 dev/urandom c 1 9
```

14. 新增 SELinux 策略，讓 dev 目錄下的所有 character device node 的 SELinux Context type 都正確。

```
[root@kvm5 script]# cd dev
[root@kvm5 dev]# ls
null  random  urandom  zero
[root@kvm5 dev]# for i in $(ls)
> do
> semanage fcontext -a -t ${i}_device_t `pwd`/$i
> done
```

15. restorecon 恢復 dev 目錄下四個 character devices 的 SELinux Contexts。

```
[root@kvm5 dev]# restorecon -v *
```

16. 查看 dev 目錄下四個 character devices 的 SELinux Contexts，已符合需求。

```
[root@kvm5 dev]# ls -Z
crw-rw-rw-. root root unconfined_u:object_r:null_device_t:s0 null
crw-rw-rw-. root root unconfined_u:object_r:random_device_t:s0 random
crw-rw-rw-. root root unconfined_u:object_r:urandom_device_t:s0 urandom
crw-rw-rw-. root root unconfined_u:object_r:zero_device_t:s0 zero
[root@kvm5 dev]# for i in $(ls)
> do semanage fcontext -a -t ${i}_device_t `pwd`/$i
> done
```

17. Linux 系統的目錄/tmp 是所有用户都可寫入讀取，且設定特殊權限 Set sticky bit。

```
[root@kvm5 dev]# cd ..
[root@kvm5 script]# chmod 1777 tmp
[root@kvm5 script]# ll -Zd tmp/
drwxrwxrwt. root root unconfined_u:object_r:tmp_t:s0   tmp/
```

18. chroot 切換根目錄到/var/www/script，進入到新建的系統

```
[root@kvm5 script]# chroot /var/www/script
bash-4.2#
```

19. 在新系統執行 touch 產生一個空檔案，因新系統中沒有 touch 指令，所以出現 command not found。

```
bash-4.2# touch aaa
bash: touch: command not found
```

20. exit 回來原系統。

```
bash-4.2# exit
exit
[root@kvm5 script]#
```

3.3 產生 SELinux 策略規則

1. 先安裝 php。

```
[root@kvm5 ~]# yum install php
```

2. php 程式要執行 chroot，所以 visudo 修改 apache 可以免密碼使用/usr/sbin/chroot。

```
[root@kvm5 html]# visudo
[root@kvm5 html]# grep apache /etc/sudoers
apache ALL=(ALL) NOPASSWD:/usr/sbin/chroot
```

3. apache 的 uid 及 gid 都是 48。

```
[root@kvm5 script]# id apache
uid=48(apache) gid=48(apache) groups=48(apache)
```

4. 在網頁根目錄/var/www/html 建立 php 測試檔 exec.php，其中 48 是 apache 的 uid 及 gid。

```
[root@kvm5 html]# vim exec.php
[root@kvm5 html]# cat exec.php
<?php
exec('sudo chroot --userspec=48:48 /var/www/script echo abc', $results);
print_r($results);
?>
```

5. 暫時關閉 SELinux。

```
[root@kvm5 html]# setenforce 0
[root@kvm5 html]# getenforce
Permissive
```

6. 備份 audit.log，並產生一個空檔 audit.log。

```
[root@kvm5 html]# mv /var/log/audit/audit.log /var/log/audit/audit.bak
[root@kvm5 html]# touch /var/log/audit/audit.log
```

7. 重新啓動 auditd 服務。

```
[root@kvm5 html]# service auditd restart
Stopping logging:                                          [  OK  ]
Redirecting start to /bin/systemctl start auditd.service
```

8. 目前 SELinux 在 Permissive 模式，php 網頁正常輸出 abc，警告資料存於/var/log/audit/audit.log。

```
[root@kvm5 html]# curl http://127.0.0.1/exec.php
Array
(
    [0] => abc
)
```

9. 將審查的安全風險記錄，轉成 SELinux 允許的策略模組。

```
[root@kvm5 html]# cat /var/log/audit/audit.log | audit2allow -M httpsudo
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i httpsudo.pp
```

10. 查看 httpsudo.te 檔，可看到必須允許的 SELinux 策略及布林值。

```
[root@kvm5 html]# cat httpsudo.te

module httpsudo 1.0;

require {
    type httpd_t;
    class capability { audit_write sys_resource };
    class process setrlimit;
    class netlink_audit_socket nlmsg_relay;
}

#============= httpd_t ==============

#!!!! This avc can be allowed using one of the these booleans:
#     httpd_run_stickshift, httpd_setrlimit
```

```
allow httpd_t self:capability { audit_write sys_resource };

#!!!! This avc can be allowed using the boolean 'httpd_mod_auth_pam'
allow httpd_t self:netlink_audit_socket nlmsg_relay;

#!!!! This avc can be allowed using the boolean 'httpd_setrlimit'
allow httpd_t self:process setrlimit;
```

11. 針對 te 檔的提示，開啓布林值 httpd_mod_auth_pam。

```
[root@kvm5 html]# setsebool httpd_mod_auth_pam 1
```

12. 啓動 SELinux。

```
[root@kvm5 html]# setenforce 1
```

13. 目前 SELinux 在 Enforcing 模式，php 網頁正常輸出 abc，表示已解決問題。

```
[root@kvm5 html]# curl http://127.0.0.1/exec.php
Array
(
    [0] => abc
)
```

3.4 自訂策略模組

1. 若開啓 te 檔提示的 SELinux 布林值還是無效，則必須增加策略模組。將上一節產生的模組加入到 SELinux 策略中。

```
[root@kvm5 html]# semodule -i httpsudo.pp
```

2. 查詢 SELinux 模組，已出現自訂的□組。

```
[root@kvm5 html]# semodule -l | grep http
httpsudo    1.0
```

3. 移除 SELinux 模組。

```
[root@kvm5 html]# semodule -r httpsudo
libsemanage.semanage_direct_remove_key: Removing last httpsudo module
(no other httpsudo module exists at another priority).
```

4. 再查詢 SELinux 模組，自訂的 httpsudo 模組已不存在。

```
[root@kvm5 html]# semodule -l | grep http
```

Chapter 4

Linux 檔案系統

4.1 簡介

1. Linux 的檔案系統格式有：Ext, Ext2, Ext3, Ext4, JFS, XFS, btrfs 及 swap。
2. Extended (Ext) 檔案系統比較如下：

檔案系統	最大檔案大小	最大檔案系統大小	檔名長度
ext			
ext2	16GiB - 2TiB	2TiB - 32TiB	255
ext3	16GiB - 2TiB	2TiB - 32TiB	255
ext4	16TiB	50TiB	255

3. Journaled File System (JFS) ：是由 IBM 開發的 64 位日誌檔案系統，支持的檔案系統大小 16MiB - 512TiB。JFS 是眞正意義上的 64 位的檔案系統，所有 JFS 檔案系統結構化欄位都是 64 位大小。
4. XFS：高效能的紀錄檔檔案系統，1993 年由 Silicon Graphics 爲 IRIX 作業系統開發。2000 年 5 月，Silicon Graphics 以 GNU 通用公眾授權條款釋出這套系統的原始碼，之後被移植到 Linux 核心上。XFS 特別擅長處理大檔案，同時提供平滑的資料傳輸。目前 CentOS 7 將 XFS+LVM 作爲預設的檔案系統。
5. B-tree file system (Btrfs)：支援寫入時複製 (copy-on-write, COW) 的檔案系統，採用 GPL 授權。改善 ext3 的限制，特別是單個檔案的大小，總檔案系統大小或檔案檢查和加入 ext3 未支援的功能，像是可寫快照 (writable snapshots)、快照的快照 (snapshots of snapshots)、內建磁碟陣列 (RAID)，以及子卷 (subvolumes)。COW：每次寫入資料時，先將資料寫入到新的 block，寫入成功後，更改舊資料塊指標到新資料塊，而非更改本身。
6. Swap：做爲實體記憶體 (RAM) 不够用時的儲存空間，當實體記憶體用完，記憶體上不活動的頁面會被移到 swap 空間。Swap 是硬碟上的檔案系統，存取速度遠比實體記憶體慢。

4.2 Linux 目錄配置

1. Linux 目錄樹配置標準參考：FHS (Filesystem Hierarchy Standard)
2. FHS 標準，主要定義三段目錄：

(a) 最上層根目錄 /

(b) 次層 /usr 及

(c) 次層 /var 的目錄內容。

3. FHS 定義的目錄資訊，主要分爲 shareable, unshareable, static, variable 四種：

	shareable	unshareable
static	/usr	/etc
	/opt	/boot
variable	/var/mail	/var/run
	/var/spool/news	/var/lock

4. / ：根目錄

```
[root@mdk-dyw /]# ls -al /
total 68
drwxr-xr-x   17 root adm   4096 Oct  5 08:17 ./
drwxr-xr-x   17 root adm   4096 Oct  5 08:17 ../
-rw-r--r--    1 root root     0 Sep 16 11:11 .autofsck
drwxr-xr-x    2 root root  4096 Sep 28 12:50 bin/
drwxr-xr-x    3 root root  4096 Sep 16 11:11 boot/
drwxr-xr-x   26 root root 14420 Oct  4 15:28 dev/
drwxr-xr-x   85 root root  8192 Oct  4 15:28 etc/
drwxr-xr-x   18 root root  4096 Sep 26 12:45 home/
drwxr-xr-x    2 root root  4096 Oct  5 08:19 initrd/
drwxr-xr-x   11 root root  4096 Sep 28 12:49 lib/
drwxr-xr-x    5 root root  4096 Oct  4 15:28 mnt/
dr-xr-xr-x  164 root root     0 Sep 16 11:10 proc/
-rw-------    1 root root  1024 Sep  9 13:06 .rnd
drwx------   16 root root  4096 Oct  5 08:19 root/
drwxr-xr-x    2 root root  8192 Sep 28 12:50 sbin/
drwxr-xr-x   10 root root     0 Sep 16 11:10 sys/
drwxrwxrwt   24 root root  4096 Oct  5 04:07 tmp/
drwxr-xr-x   12 root root  4096 Sep  9 13:15 usr/
drwxr-xr-x   23 root root  4096 Sep  9 17:31 var/
```

5. 各目錄大致內容：

/	根目錄，/etc, /bin, /dev, /lib, /sbin 這五個次目錄都要與根
/bin	放常用執行檔如：ls, mv, rm。
/boot	放 Linux 核心及開機相關檔案。
/boot/grub	放 grub 開機管理程式。
/dev	放裝置有關的檔案，Linux 把裝置當成檔案看待。
/etc	放系統在開機過程需要讀取的檔案。
/etc/rc.d	記錄開關機過程中的 scripts 檔案。
/etc/init.d	放所有服務預設的啓動 scripts。
/etc/xinetd.d	放一些額外的服務。
/etc/X11	放 X window 有關的設定檔。
/home	使用者的家目錄。
/lib	放 Linux 執行或編譯程式時要使用的函式庫 (library)。
/mnt	軟碟及光碟預設掛載的地方。
/proc	虛擬檔案系統，不佔任何硬碟空間，放置的資料在記憶體
/root	系統管理員的家目錄。
/sbin	放系統管理用的程式如：fdisk, mount。
/sys	開機核心載入的驅動及模組。
/tmp	用户暫時存放檔案的目錄。
/usr	此目錄包括許多子目錄，用來存放系統指令、安裝程式及
/usr/bin	一般使用者可執行的程式執行檔。
/usr/include	套件的 header 檔。
/usr/lib	內含許多程式與子程式所需的函式庫。
/usr/local	用户額外安裝的程式可放此目錄，以區分原始系統安裝的
/usr/man	放程式的說明檔。
/usr/sbin	安裝套件中系統管理員可用之程式。
/usr/src	原始碼。
/usr/share	安裝的程式及套件。
/usr/doc	程式的說明文件。
/usr/man	程式的 man 手冊。
/var	系統執行中，常態性變動的檔案。
/var/cache	程式執行中的暫存檔。
/var/lib	程式執行中需要使用的資料檔案，例如 rpm 資料庫系統。
/var/log	日誌檔。
/var/run	某些程式或服務啓動後會將其 PID 放置於此。
/var/spool	一些佇列資料，例如：主機收到電子郵件後，放於/var/spo

4.3 目錄與路徑

1. 相對路徑與絕對路徑。

   (a) 絕對路徑：從根目錄 / 寫起，如：目錄 /usr/share/man。

   (b) 相對路徑：不是從根目錄 / 寫起，其寫法如：

.	代表此層目錄
..	代表上層目錄
–	代表前一個工作目錄
~	代表自己的家目錄
~user	代表到 user 這個用户的家目錄

2. 相關指令

cd	變換工作目錄。
pwd	顯示目前所在的目錄。
mkdir	建立新目錄。
rmdir	刪除【空】的目錄。

3. cd (Change Directory)：變換工作目錄。

```
[root@dywHome2 ~]# cd ..
[root@dywHome2 /]# cd
[root@dywHome2 ~]# cd /home/dywang
[root@dywHome2 dywang]# cd ~
[root@dywHome2 ~]# cd ~dywang
[root@dywHome2 dywang]# cd ../../var
[root@dywHome2 var]# cd ~test
-bash: cd: ~test: No such file or directory
```

4. pwd (Print Working Directory)：顯示目前所在的目錄。

```
[root@dywHome2 var]# cd /var/log
[root@dywHome2 log]# pwd
/var/log
```

5. mkdir (Make Directory)：建立新目錄。

```
[root @test /root ]# mkdir [-mp] 目錄名稱[]參數說明：

-m ：直接設定檔案的權限，而不管 umask 內容。
-p ：直接將上層目錄遞迴建立起來。
#範例：
[root @test /root]# cd tmp
[root @test /tmp]# mkdir test <== 建立名稱爲 test 的目錄。
[root @test /tmp]# mkdir -p test1/test2/test3/test4 <== 直接建立
   test2... 等上層目錄。
[root @test /tmp]# mkdir -m 711 testqq  <== 建立權限爲 711 的目錄。
[root @test /tmp]# ll test*
drwxrwxr-x    2 test    test        4096 Feb  6 20:47 test/
drwxrwxr-x    3 test    test        4096 Feb  6 20:48 test1/
drwx--x--x    2 test    test        4096 Feb  6 20:48 testqq/
```

6. rmdir (Remove Directory)：刪除【空】的目錄。

```
[root @test /root ]# rmdir [-p] 目錄名稱[]參數說明：

-p ：將上層的目錄也刪除。
#範例：
[root @test /root]# rmdir test <== 刪除名稱爲 test 的目錄
[root @test tmp]# ll
```

```
drwxrwxr-x    3 test    test        4096 Feb  6 20:48 test1/
[root @test tmp]# rmdir test1
rmdir: `test1': Directory not empty
[root @test tmp]# rmdir -p test1/test2/test3/test4
[root @test tmp]$ ll
```

7. 環境變數 $PATH：提供搜尋指令之路徑

   (a) 列出環境變數 PATH：

```
[root@mdk-dyw ~]# echo $PATH
/sbin:/usr/sbin:/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:/usr/
    local/sbin
```

   (b) 若目前工作目錄 /home/csie 下有一可執行檔 foo，但本目錄並不在 $PATH 設定中，如何執行 foo？

```
[root@mdk-dyw ~]# /home/csie/foo
[root@mdk-dyw ~]# ./foo
```

4.4 檔案權限

1. 檔案權限與屬性為學習 Linux 之重要關卡，在根目錄 / 下輸入指令 ls -al 顯示如下：

```
[dywang@mdk-dyw /]$ ls -al
total 72
drwxr-xr-x   18 root adm    4096 Sep 16 11:11 ./
drwxr-xr-x   18 root adm    4096 Sep 16 11:11 ../
-rw-r--r--    1 root root      0 Sep 16 11:11 .autofsck
drwxr-xr-x    2 root root   4096 Sep 28 12:50 bin/
drwxr-xr-x    3 root root   4096 Sep 16 11:11 boot/
drwxr-xr-x   26 root root  14420 Oct  1 12:30 dev/
drwxr-xr-x   85 root root   8192 Sep 29 12:59 etc/
drwxr-xr-x   18 root root   4096 Sep 26 12:45 home/
drwxr-xr-x    2 root root   4096 Sep  9 16:14 initrd/
drwxr-xr-x   11 root root   4096 Sep 28 12:49 lib/
drwxr-xr-x    5 root root   4096 Sep 29 12:59 mnt/
drwxr-xr-x    2 root root   4096 Jan  5  2004 opt/
dr-xr-xr-x  179 root root      0 Sep 16 11:10 proc/
-rw-------    1 root root   1024 Sep  9 13:06 .rnd
drwx------   16 root root   4096 Oct  1 13:13 root/
drwxr-xr-x    2 root root   8192 Sep 28 12:50 sbin/
drwxr-xr-x   10 root root      0 Sep 16 11:10 sys/
drwxrwxrwt   23 root root   4096 Oct  1 13:13 tmp/
drwxr-xr-x   12 root root   4096 Sep  9 13:15 usr/
```

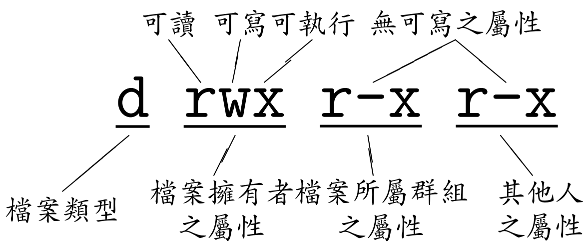
```
drwxr-xr-x   23 root root  4096 Sep  9 17:31 var/
```

2. total 72：檔案共 72 Blocks。

3. 各欄位說明：

欄位一	欄位二	欄位三	欄位四	欄位五	欄位六	欄位七
drwxr-xr-x	3	root	root	4096	Sep 16 11:11	boot
檔案屬性	硬連結數目	擁有者	所有者群組	大小	建檔日期	檔名

4. 第一欄位 `drwxr-xr-x`，共有檔案的 10 個屬性：



5. 第一屬性代表這個檔案是『目錄、檔案或連結檔』:
   (a) [d] 是目錄；
   (b) [-] 是檔案；
   (c) [l] 爲連結檔 (link file)；
   (d) [b] 表示爲裝置檔裡面的可供儲存的周邊設備，例如硬碟；
   (e) [c] 表示爲裝置檔裡面的序列埠設備，例如鍵盤、滑鼠。

6. 接下來的屬性中，三個爲一組，第一組爲『擁有人的權限』; 第二組爲『同群組的權限』; 第三組爲『其他非本群組的權限』，且均爲『rwx』的三個參數的組合。其中，
   [r] 代表可讀、
   [w] 代表可寫、
   [x] 代表可執行、
   [-] 代表無該權限。

7. 對於目錄光有可讀權限 [r] 並無法存取目錄，必須要有權限 [x]，才能執行 `ls`, `cd` 等指令，以進入目錄。

8. Linux 檔案是否可執行與附檔名無關，而是其有無可執行 [x] 的權限。

9. `drwxr-xr-x` 代表
   (a) 爲一目錄；
   (b) 檔案擁有者可讀、可寫亦可執行；
   (c) 檔案所屬群組可讀、可執行但不可寫；
   (d) 其他人可讀、可執行但不可寫。

10. 第二欄表示爲連結佔用的節點 (i-node) (若爲目錄時，通常與該目錄底下還有多少目錄有關) 這部分將在介紹連結 link 檔案時 (下一節) 再深入的介紹；

11. 第三欄表示這個檔案（或目錄）的『擁有人』；
12. 第四欄表示擁有人的群組；
13. 第五欄爲這個檔案的大小；
14. 第六欄爲這個檔案的建檔日期或者是最近的修改日期，分別爲月份、日期及時間。
15. 第七欄爲這個檔案的檔名，如果檔名之前多一個『.』，則代表這個檔案爲『隱藏檔』。

4.5 檔案格式與種類

1. 檔案種類
   (a) 正規檔案 [-]：
      i. 純文字檔 (ascii)；
      ii. 二進位檔 (binary)；
      iii. 資料格式檔 (data)，例如/var/log/wtmp。
   (b) 目錄 [d]
   (c) 連結檔 [l]：類似 windows 之捷徑。
   (d) 設備檔 [b,c,s,p]：集中於/dev 目錄下。
      i. 區塊 (block) 設備檔 [b]：儲存資料如硬碟。
      ii. 字元 (character) 設備檔 [c]：序列埠的周邊設備如鍵盤。
      iii. 資料接口檔 (sockets) [s]：用在網路上的資料承接 (/var/run)。
      iv. 資料輸送檔 (FIFO,pipe) [p]：解決多個程序同時存取一個檔案所造成的問題。
2. Linux 檔案附檔名
   (a) Linux 檔案附檔名爲個人視別用，並不具實質作用。
   (b) Linux 檔案能否執行要視其屬性，與檔名及附檔名無關。
   (c) Linux 每個檔案或目錄的檔名最長可以達 255 個字元。
   (d) Linux 常用之附檔名
      i. Shell 腳本 (shell scripts)：通常以 *.sh 代表。
      ii. 打包或壓縮檔：通常附檔名爲 *.Z, *.tar, *.tar.gz, *.zip, *.tgz 等等。
      iii. 網頁相關檔案：通常使用 *.html 與 *.php 等

4.6 檔案權限例子說明

1. 進入某目錄成爲『可工作目錄』的基本權限
   (a) 使用者可進入該目錄需具備 x 權限

(b) 使用者可在該目錄查閱檔名，需具備 r 權限

2. 讀取一個檔案的基本權限
   (a) 使用者在該檔案所在的目錄至少要有 x 權限
   (b) 使用者對該檔案至少要有 r 權限
3. 修改一個檔案的基本權限
   (a) 使用者在該檔案所在的目錄至少要有 x 權限
   (b) 使用者對該檔案至少要有 r, w 權限
4. 建立一個檔案的基本權限
   (a) 使用者在該目錄要具有 w,x 的權限
5. 進入某目錄並執行該目錄下的某個指令
   (a) 使用者在該目錄至少要有 x 的權限
   (b) 使用者在該檔案至少需要有 x 的權限
6. 例題：在 Linux 檔案系統中，刪除一個檔案最少具備什麼權限？A
   (A) 只需對檔案所在目錄具有 wx
   (B) 只需對檔案所在目錄具有 wx，及對檔案具有 w
   (C) 只需對檔案所在目錄具有 wx，及對檔案具有 rw
   (D) 只需對檔案所在目錄具有 w ，及對檔案具有 w
7. 例題：使用者 foo 屬於群組 foo，有一個目錄名稱爲 foo，屬於使用者 root 及群組 root，並且目錄 foo 的存取權限爲『`drwxr-x--x`』，請問下列哪些正確？AD
   (A) 使用者 foo 可以進入目錄 foo
   (B) 使用者 foo 不能進入目錄 foo
   (C) 使用者 foo 可以由 ls 看到目錄 foo 下有哪些檔案
   (D) 如果目錄 foo 中有一檔案 bar 其存取權限爲 777，使用者 foo 可以讀取 bar

Chapter 5

Linux 檔案存取管理

5.1 使用者與群組

1. Linux 爲多人多工系統。
2. 檔案擁有者保有隱私權。
3. 檔案所屬群組 --團隊功能。

5.2 檔案權限管理命令

1. chgrp：改變檔案所屬群組，群組名稱必須存在，否則會產生錯誤。

   chgrp 群組名稱 檔案或目錄

2. chown：改變檔案擁有者及所屬群組，帳號名稱必須存在，否則會產生錯誤。選項 [-R] 可以連目錄下的所有次目錄或檔案一起更改。

```
chown [ -R ] 帳號名稱檔案或目錄
chown [ -R ] 帳號名稱群組名稱: 檔案或目錄
```

3. chmod：改變權限。

```
chmod [ -R ] xyz 檔案或目錄
chmod [ -R ] u = r 檔案或目錄
             g + w 檔案或目錄
             o - x 檔案或目錄
             a 檔案或目錄
chmod u=rwx,go=rx 檔案或目錄
chmod a-x 檔案或目錄
```

(a) xyz 爲三組 rwx 屬性數值之加總。
r : 4
w : 2
x : 1

(b) -rwxr-xr-- 之屬性數值爲 754。

(c) 符號類型
u : user
g : group
o : other
a : all
+ : 加入
- : 除去
= : 設定

4. 例題：若從光碟複製某一目錄 abc （包含次目錄之所有檔案）至硬碟，則目前使用者雖爲檔案擁有者，但卻無法修改該檔案，請問如何以簡單的方式，讓使用者可以修改所有檔案？

```
chmod u+w -R abc
```

5.3 檔案與目錄管理

1. 常用指令

ls	顯示檔案名稱、屬性等。
cp	拷貝檔案或目錄。
rm	移除檔案或目錄。
mv	檔案移動或改名。
basename	移除前導目錄及字尾，顯示檔名。
dirname	移除檔名，顯示目錄。

2. ls (list directory contents)：顯示檔案名稱、屬性等。

```
[root @test /root ]# ls [-ailS]參數說明：

-a 全部的檔案都列出（連同隱藏檔）      :
-i 印出      : inode 的值
-l 長的列出，連同檔案大小的資料等等      :
-S 以檔案大小排序      :
--color=never 不要顯示顏色      :
--color=always 均顯示顏色      :
--color=auto 由系統自行判斷      :
```

3. ls 範例：

```
[root@dywHome2 ~]# ls
col.txt  drakx/  echoname*  findnewer  sed  tmp/
[root@dywHome2 ~]# ls -a
./                 findnewer          .kde.bak/  .tcshrc
../                .gconf/            .lesshst   .test.txt.swp
.bash_completion  .gconfd/            .local/    tmp/
.bash_history      .gstreamer-0.8/    .mc/       .viminfo
.bash_logout       .gtk-bookmarks     .mcop/     .vimrc
.bash_profile      .gtk_qt_engine_rc  .mcoprc    .xauth/
.bashrc            .gtkrc*            .mplayer/  .Xdefaults
col.txt            .gtkrc-2.0*        .qt/       .xdg_menu_cache/
.cshrc             .gtkrc-2.0.bak*    .rpmdrake
drakx/             .ICEauthority      sed
echoname*          .kde/              .ssh/
[root@dywHome2 ~]# ls -l
total 6
-rw-r--r-- 1 root root 2939 Feb  2  2008 col.txt
drwx------ 2 root root 1024 Dec 27  2007 drakx/
-rwxr-xr-x 1 root root   11 Jan 28  2008 echoname*
-rw-r--r-- 1 root root    0 Aug 30 08:44 findnewer
-rw-r--r-- 1 root root    0 Apr 16 09:35 sed
drwx------ 5 root root 1024 Sep  3 19:59 tmp/
[root@dywHome2 ~]# ls -al
total 90
drwx------ 17 root root  2048 Sep  3 20:33 ./
drwxr-xr-x 21 root adm   1024 Sep  3 19:59 ../
-rw-r--r--  1 root root   145 Aug  4  2006 .bash_completion
-rw-------  1 root root 23796 Sep  3 17:47 .bash_history
-rw-r--r--  1 root root    24 Aug  4  2006 .bash_logout
-rw-r--r--  1 root root   106 Aug  4  2006 .bash_profile
-rw-r--r--  1 root root   226 Aug  4  2006 .bashrc
-rw-r--r--  1 root root  2939 Feb  2  2008 col.txt
-rw-r--r--  1 root root   233 Aug  4  2006 .cshrc
drwx------  2 root root  1024 Dec 27  2007 drakx/
### 中間省略 ###
-rw-r--r--  1 root root  1479 Aug  4  2006 .Xdefaults
drwxr-xr-x  2 root root  1024 Apr 20 10:45 .xdg_menu_cache/
[root@dywHome2 ~]# ls -al | more
```

4. cp (copy files and directories)：拷貝檔案或目錄。

```
[root@dywHome2 ~]# cp [-dLprRsufa] 來源檔[] 目的檔[]參數說明：

-d ：連結檔維持其路徑之連結
-L ：解除連結，直接      copy 連結的檔案（預設）
-p ：維持檔案之屬性
-r, -R   ：連同目錄下之所有檔案及次目錄都 copy
-s ：做成連結檔，而不，與      copy ln 指令相同功能
-u, --：來源檔較新才會進行update copy 的動作，常用於備份的動作
-f ：強制，不再尋問是否覆蓋      copy
-a ：與      -dpR 相同
```

5. mkdir 範例：

```
[root@dywHome2 ~]# mkdir test
[root@dywHome2 ~]# cd test
[root@dywHome2 test]# touch foo
[root@dywHome2 test]# cp foo foo1
[root@dywHome2 test]# ln -s foo lnfoo
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root root 0 Sep  3 20:50 foo
-rw-r--r-- 1 root root 0 Sep  3 20:50 foo1
lrwxrwxrwx 1 root root 3 Sep  3 20:52 lnfoo -> foo
[root@dywHome2 test]# cp lnfoo foo2
[root@dywHome2 test]# cp -d lnfoo foo3
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root root 0 Sep  3 21:03 foo
-rw-r--r-- 1 root root 0 Sep  3 20:50 foo1
-rw-r--r-- 1 root root 0 Sep  3 20:52 foo2
lrwxrwxrwx 1 root root 3 Sep  3 21:11 foo3 -> foo
lrwxrwxrwx 1 root root 3 Sep  3 20:52 lnfoo -> foo
[root@dywHome2 test]# chown dywang:users foo2
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root   root  0 Sep  3 21:03 foo
-rw-r--r-- 1 root   root  0 Sep  3 20:50 foo1
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo2
lrwxrwxrwx 1 root   root  3 Sep  3 21:11 foo3 -> foo
lrwxrwxrwx 1 root   root  3 Sep  3 20:52 lnfoo -> foo
[root@dywHome2 test]# cp foo2 foo4
[root@dywHome2 test]# cp -p foo2 foo5
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root   root  0 Sep  3 21:03 foo
-rw-r--r-- 1 root   root  0 Sep  3 20:50 foo1
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo2
lrwxrwxrwx 1 root   root  3 Sep  3 21:11 foo3 -> foo
-rw-r--r-- 1 root   root  0 Sep  3 21:12 foo4
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo5
lrwxrwxrwx 1 root   root  3 Sep  3 20:52 lnfoo -> foo
[root@dywHome2 test]# cd ..
[root@dywHome2 ~]# cp -a test test1
[root@dywHome2 ~]# ll test test1
test:
total 0
-rw-r--r-- 1 root   root  0 Sep  3 21:03 foo
-rw-r--r-- 1 root   root  0 Sep  3 20:50 foo1
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo2
lrwxrwxrwx 1 root   root  3 Sep  3 21:11 foo3 -> foo
-rw-r--r-- 1 root   root  0 Sep  3 21:12 foo4
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo5
```

```
lrwxrwxrwx 1 root   root  3 Sep  3 20:52 lnfoo -> foo

test1:
total 0
-rw-r--r-- 1 root   root  0 Sep  3 21:03 foo
-rw-r--r-- 1 root   root  0 Sep  3 20:50 foo1
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo2
lrwxrwxrwx 1 root   root  3 Sep  3 21:13 foo3 -> foo
-rw-r--r-- 1 root   root  0 Sep  3 21:12 foo4
-rw-r--r-- 1 dywang users 0 Sep  3 20:52 foo5
lrwxrwxrwx 1 root   root  3 Sep  3 21:13 lnfoo -> foo
```

6. rm (remove files or directories)：移除檔案或目錄。

```
[root@dywHome2 ~]# rm [-fir] 檔名[]參數說明：

-i 提供使用者確認（預設值）      :
-r,-R 循環，含次目錄 :
-f 忽略不存在的檔，強制移除不提示。     :
```

7. rm 範例：

```
[root@dywHome2 ~]# rm test/foo1
rm: remove regular empty file `test/foo1'? y
[root@dywHome2 ~]# rm -rf test
```

8. mv (move (rename) files)：檔案移動或改名。

```
[root@dywHome2 ~]#  mv [-u] 來源檔[] 目的檔[]參數說明：

-u,--updates 來源檔比目的檔新時才動作:
```

9. mv 範例：

```
[root@dywHome2 ~]# mkdir test
[root@dywHome2 ~]# cd test
[root@dywHome2 test]# touch foo
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root root 0 Sep  3 21:25 foo
[root@dywHome2 test]# mv foo foo1
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root root 0 Sep  3 21:25 foo1
[root@dywHome2 test]# cp foo1 foo2
```

```
[root@dywHome2 test]# ll
total 0
-rw-r--r-- 1 root root 0 Sep  3 21:25 foo1
-rw-r--r-- 1 root root 0 Sep  3 21:26 foo2
[root@dywHome2 test]# mv foo1 foo2
mv: overwrite `foo2'? n
[root@dywHome2 test]# mv -u foo1 foo2
[root@dywHome2 test]# mv foo1 foo2 /tmp
[root@dywHome2 test]# ll /tmp/foo[12]
-rw-r--r-- 1 root root 0 Sep  3 21:25 /tmp/foo1
-rw-r--r-- 1 root root 0 Sep  3 21:26 /tmp/foo2
```

10. basename：移除前導目錄及字尾，顯示檔名。

```
[root@dywHome2 test]#  basename NAME 字尾[]
```

11. basename 範例：

```
[root@dywHome2 test]# basename /root/test/foo.sh
foo.sh
[root@dywHome2 test]# basename /root/test/foo.sha
foo.sha
[root@dywHome2 test]# basename /root/test/foo.sh .sh
foo
[root@dywHome2 test]# basename /root/test/foo.sh o.sh
fo
```

12. dirname：移除檔名，顯示目錄。

```
[root @test /root ]# dirname NAME
```

13. dirname 範例：

```
[root@dywHome2 test]# dirname /root/test/foo.sh
/root/test
[root@dywHome2 test]# dirname /root/test/
/root
[root@dywHome2 test]# dirname foo.sh
.  <== foo.sh 沒有前導目錄，故輸出爲. 表示目前目錄。
```

Chapter 6

隱藏權限及特殊屬性

6.1 隱藏權限

1. 相關指令
   umask　查詢及設定預設權限。
   chattr　改變檔案的隱藏屬性。
   lsattr　顯示檔案的隱藏屬性。

2. 檔案的預設權限

   (a) 若 umask 爲 022，則所屬群組及其他使用者皆須扣除『寫』的權限。
      i. 建立檔案時：(-rw-rw-rw-) - (-----w--w-) ==> -rw-r--r--
      ii. 建立目錄時：(drwxrwxrwx) - (-----w--w-) ==> drwxr-xr-x

   (b) umask：查詢及設定預設權限。

```
[root@dywHome2 test]# umask [-S] [mode]參數說明：

-S(Symbolic): 以符號類型顯示權限
```

   (c) umask 範例

```
[root@dywHome2 test]# umask
0022
[root@dywHome2 test]# umask -S
u=rwx,g=rx,o=rx
[root@dywHome2 test]# touch foo1
[root@dywHome2 test]# mkdir test1
[root@dywHome2 test]# ll
total 1
-rw-r--r-- 1 root root    0 Sep  4 10:53 foo1
drwxr-xr-x 2 root root 1024 Sep  4 10:54 test1/
[root@dywHome2 test]# umask 002
[root@dywHome2 test]# umask u=rwx.g=rwx,o=rx
[root@dywHome2 test]# touch foo2
[root@dywHome2 test]# mkdir test2
[root@dywHome2 test]# ll
total 2
```

```
-rw-r--r-- 1 root root    0 Sep  4 10:53 foo1
-rw-rw-r-- 1 root root    0 Sep  4 10:54 foo2
drwxr-xr-x 2 root root 1024 Sep  4 10:54 test1/
drwxrwxr-x 2 root root 1024 Sep  4 10:54 test2/
```

3. 檔案的隱藏屬性

(a) chattr：改變檔案的隱藏屬性。

```
[root@dywHome2 test]# chattr +-=[ASacDdIijsTtu] 檔案或目錄名稱[]參
    數說明：  ：分別爲

+-= [+ 增加] [- 減少] [= 設定] 屬性。    ：存取時間
A atime (access) 不可被修改。(驗證未成功)     ：類似
S sync 的功能，會將資料同步寫入磁碟中。    ：檔案只能增加資料，而不
    能刪除，只有
a root 才能設定這個屬性。 ###    ：自動將檔案『壓縮』，讀取時會自動
    解壓縮。
c   ：當
d dump 備份()程序被執行時，設定 d 屬性將可使該檔案或目錄()具
    有 dump 功效。    ：檔案不能被刪除、改名、設定連結也無法寫入或新
    增資料。
i ###    ：檔案在寫入時先記錄在
j journal 中。    ：檔案會被完全的移除出硬碟空間。
s    ：與
u s 相反，資料內容還存在磁碟中，可以。 undeletion

The letters 'acdijsuADST' select the  new  attributes  for  the
    files:
append only (a), compressed (c), no dump (d), immutable (i), data
    jour-
nalling (j), secure deletion (s), no tail-merging (t), undeletable
    (u),
no  atime  updates  (A), synchronous directory updates (D),
    synchronous
updates (S), and top of directory hierarchy (T).
```

(b) chattr 範例

```
[root@dywHome2 test]# chattr +a foo
[root@dywHome2 test]# echo append > foo
-bash: foo: Operation not permitted
[root@dywHome2 test]# echo append >> foo
[root@dywHome2 test]# cat foo
append
[root@dywHome2 test]# chattr +i foo
[root@dywHome2 test]# mv foo foo1
mv: cannot move `foo' to `foo1': Operation not permitted
[root@dywHome2 test]# chattr -i foo
[root@dywHome2 test]# mv foo foo1
```

(c) `lsattr`：顯示檔案的隱藏屬性。

```
[root@dywHome2 test]# lsattr [-adR]參數說明：

-   ：列出所有檔案，含隱藏檔；a
-   ：只列出目錄，而不列目錄內容；d
-   ：連同子目錄一併列出。R
```

(d) `lsattr` ext3 檔案系統範例

```
[root@dywHome2 test]# lsattr
----ia------- ./foo
------------- ./test1
------------- ./foo2
------------- ./test2
```

(e) `lsattr` ext4 檔案系統範例

```
[root@kvm5 ~]# lsattr
-------------e-- ./anaconda-ks.cfg
-------------e-- ./resolv.conf
-------------e-- ./httpsudo.pp
-------------e-- ./httpsudo.te
```

6.2 檔案的特殊屬性 SUID/SGID/SBIT

1. Set UID (SUID)；設置使文件在執行階段具有文件所有者的權限。
   (a) SUID 典型的的例子爲/etc/passwd。
      i. 帳號與密碼存放於/etc/passwd 與/etc/shadow。而/etc/shadow 的權限爲『-r--------』，且擁有者爲 root。
      ii. 一般身份使用者要更新自己的密碼，執行/usr/bin/passwd 卻可以存取密碼檔/etc/shadow。
   (b) SUID 僅可用在『二進位制檔案 (binary file)』上。
2. Set GID (SGID)：只對目錄有效。
   (a) 目錄被設置後，任何用戶在此目錄下建立的文件，所屬群組皆與該目錄所屬的群組相同。
   (b) 可使用在特定的多人團隊的專案開發上。
3. Sticky Bit (SBIT)：防刪除屬性，只對目錄有效。
   (a) 在具有 SBIT 的目錄下，使用者若在該目錄下具有 w 及 x 的權限，則當使用者在該目錄下建立檔案或目錄時，只有檔案擁有者與 root 才有權力刪除。

(b) 典型的例子如/tmp 的權限是『drwxrwxrwt』，任何用户都可在/tmp 內新增、修改檔案，但僅有該檔案/目錄建立者與 root 能夠刪除自己的目錄或檔案。

4. SUID/SGID/SBIT 權限的顯示與設定

(a) 使用 ls -l 列出檔案屬性，如：

rwsrw-r--	表示設置 SUID 屬性
rwxrwsrw-	表示設置 SGID 屬性
rwxrw-rwt	表示設置 SBIT 屬性

(b) 如果本來在該位上有 x 屬性，則這些特殊屬性顯示爲小寫字母 (s, s, t)，否則顯示爲大寫字母 (S, S, T)。

(c) SUID/SGID/SBIT 權限設定與一般 rwx 屬性之設定類似：
4 爲 SUID
2 爲 SGID
1 爲 SBIT

(d) 例題：更改並顯示檔案 foo 的屬性。

```
[root@dywHome2 test]# chmod 4755 foo; ll foo
-rwsr-xr-x 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 6755 foo; ll foo
-rwsr-sr-x 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 1755 foo; ll foo
-rwxr-xr-t 1 root root 7 Sep  4 13:25 foo*
[root@dywHome2 test]# chmod 7666 foo; ll foo
-rwSrwSrwT 1 root root 7 Sep  4 13:25 foo
```

(e) Linux 系統中典型的例子：

```
[root@dywHome2 test]# ll /usr/bin/passwd; ls -ld /var/spool/mail;\
>ls -ld /tmp
-r-s--x--x 1 root root 15552 May  8  2006 /usr/bin/passwd*
drwxrwsr-x 2 root mail 1024 Aug  3  2006 /var/spool/mail/
drwxrwxrwt 26 root root 2048 Sep  4 15:51 /tmp/
```

6.3 特殊屬性 SGID 練習

1. 先建立練習用目錄/home/shared

```
[root@kvm7 ~]# mkdir /home/shared
```

2. 查看目錄/home/shared，目前擁有者及群組皆爲 root，擁有者可讀、可寫、可訪問此目錄，同群組及其他人只能訪問及讀取。

```
[root@kvm7 ~]# ll -d /home/shared
drwxr-xr-x. 2 root root 1024 Nov 17 19:34 /home/shared
```

3. 改變目錄/home/shared 群組爲 sharegrp，如果 sharegrp 群組不存在，必須先新增。

```
[root@kvm7 ~]# chgrp sharegrp /home/shared
```

4. 查看目錄/home/shared 群組爲 sharegrp。

```
[root@kvm7 ~]# ll -d /home/shared
drwxr-xr-x. 2 root sharegrp 1024 Nov 17 19:34 /home/shared
```

5. 改變目錄/home/shared，群組可讀、可寫、可訪問此目錄，其他人沒有任何權限，且設定 SGID，讓用户在此目錄內產生的檔案，所屬群組皆爲 sharegrp。

```
[root@kvm7 ~]# chmod 2770 /home/shared
```

6. 再查看目錄/home/shared，其屬性出現 drwxrws---，符合要求。

```
[root@kvm7 ~]# ll -d /home/shared
drwxrws---. 2 root sharegrp 1024 Nov 17 19:34 /home/shared
```

6.4 特殊屬性 SGID 測試

1. 切換身份成 deyu1。

```
[root@kvm7 ~]# su - deyu1
Last login: Sat Nov  3 02:50:16 CST 2018 on pts/0
```

2. deyu1 的附屬群組是 sharegrp，所以可以進到 SGID 的目錄/home/shared。

```
[deyu1@kvm7 ~]$ cd /home/shared/
```

3. 查看目錄/home/shared 中沒有任何檔案。

```
[deyu1@kvm7 shared]$ ll
total 0
```

4. 新增空檔案 deyu1.txt。

```
[deyu1@kvm7 shared]$ touch deyu1.txt
```

5. 查看新增的空檔案 deyu1.txt，擁有者是 deyu1，因為目錄/home/shared SGID，所以在此目錄中新增的檔案群組是 sharegrp。

```
[deyu1@kvm7 shared]$ ll
total 1
-rw-rw-r--. 1 deyu1 sharegrp 0 Nov 23 14:11 deyu1.txt
```

6. 退出帳號 deyu1，再切換成 deyu2。

```
[deyu1@kvm7 shared]$ exit
logout
[root@kvm7 ~]# su - deyu2
```

7. deyu2 的附屬群組也是 sharegrp，所以也可以進到 SGID 的目錄/home/shared。

```
[deyu2@kvm7 ~]$ cd /home/shared/
```

8. 目錄/home/shared 中同群組的帳號，都可以讀取、寫入，所以 deyu2 也可以變更 deyu1 產生的檔案 deyu1.txt。

```
[deyu2@kvm7 shared]$ echo 'abcde' >> deyu1.txt
[deyu2@kvm7 shared]$ cat deyu1.txt
abcde
```

9. 退出帳號 deyu2，再切換成 deyu4。

```
[deyu2@kvm7 shared]$ exit
logout
[root@kvm7 ~]# su - deyu4
```

10. deyu4 不是 sharegrp 群組成員，所以無法進入目錄/home/shared。

```
[deyu4@kvm7 ~]$ cd /home/shared/
-bash: cd: /home/shared/: Permission denied
```

11. 退出 deyu4 帳號。

```
[deyu4@kvm7 ~]$ exit
logout
```

Chapter 7

Access Control Lists

7.1 簡介

1. 什麼是 ACL (Access Control List)？
   (a) 提供傳統的 owner,group,others 的 read,write,execute 權限之外的細部權限設定。
   (b) ACL 可以針對單一使用者，單一檔案或目錄來進行 r, w, x 的權限規範。
   (c) ACL 主要可以針對以下項目來控制權限：
      i. 使用者 (user)：可以針對使用者來設定權限；
      ii. 群組 (group)：針對群組爲對象來設定其權限；
      iii. 預設屬性 (mask)：針對在該目錄下在建立新檔案/目錄時，規範新資料的預設權限。

2. setfacl：設定 ACL 的狀態。

```
[root@demo ~]# setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file
    ] file ...選
    項：

-m ：設定後續的 acl 參數給檔案使用，不可與 -x 合用；
-x ：刪除後續的 acl 參數，不可與 -m 合用；
-b ：移除所有的 ACL 設定參數；
-k ：移除預設的 ACL 參數；
-R ：遞迴設定，即包括次目錄都會被設定； acl
-d ：設定『預設 acl 參數』，只對目錄有效，在該目錄新建的資料會引用此預設
    值
acl_spec:
[d[efault]:] [u[ser]:]uid [:perms]
[d[efault]:] g[roup]:gid [:perms]
[d[efault]:] m[ask][:] [:perms]
[d[efault]:] o[ther][:] [:perms]
# d 選項，表示該目錄預設有設定的 acl 規範，類似 umask 的功能。
```

7.2 ACL 練習

1. 先複製測試檔案

```
[root@kvm7 ~]# cp /etc/fstab /var/tmp/
```

2. 查看複製的檔案擁有者，群組皆爲 root，目前權限爲 644，root 有讀取及寫入的權限。

```
[root@kvm7 ~]# ll /var/tmp/fstab
-rw-r--r--. 1 root root 694 Oct 15 06:29 /var/tmp/fstab
```

3. 使用 getfacl 查看/var/tmp/fstab 目前的 ACL，沒有其他用户有特殊的權限。

```
[root@kvm7 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

4. 改變/var/tmp/fstab 的 ACL，用户 deyu1 有讀取及寫入的權限。

```
[root@kvm7 ~]# setfacl -m u:deyu1:rw /var/tmp/fstab
```

5. 改變/var/tmp/fstab 的 ACL，用户 deyu2 沒有任何權限。

```
[root@kvm7 ~]# setfacl -m u:deyu2:- /var/tmp/fstab
```

6. 再使用 getfacl 查看/var/tmp/fstab 目前的 ACL，其他用户 deyu1 及 deyu2 有特殊的權限。

```
[root@kvm7 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu1:rw-
user:deyu2:---
group::r--
mask::rw-
other::r--
```

7. 列出/var/tmp/fstab 可以看到權限部分多了個 +，且權限不是原本的“644”？

```
[root@kvm7 ~]# ll /var/tmp/fstab
-rw-rw-r--+ 1 root root 694 Oct 15 06:29 /var/tmp/fstab
```

7.3 ACL 變更

1. 測試檔案/var/tmp/fstab 目前的 ACL。

```
[root@kvm7 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

2. /var/tmp/fstab 新增 deyu4 可讀寫。

```
[root@kvm7 ~]# setfacl -m u:deyu4:rw /var/tmp/fstab
```

3. 查詢/var/tmp/fstab 的 ACL 有 deyu4 可讀寫。

```
[root@kvm7 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu4:rw-
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

4. 刪除/var/tmp/fstab ACL 中的 deyu4 可讀寫。

```
[root@kvm7 ~]# setfacl -x u:deyu4 /var/tmp/fstab
```

5. 查詢/var/tmp/fstab 的 ACL 沒有 deyu4 設定了。

```
[root@kvm7 ~]# getfacl /var/tmp/fstab
getfacl: Removing leading '/' from absolute path names
# file: var/tmp/fstab
# owner: root
# group: root
user::rw-
user:deyu2:---
user:deyu1:rw-
group::r--
mask::rw-
other::r--
```

Chapter 8

Rsync 檔案備援

8.1 前言

1. 參考資料
2. rsync 是 Linux 系統上常用來複製與備份檔案的工具，可以處理本機或遠端的檔案同步工作。
3. rsync 可以同步同一台電腦上的檔案系統，或同步異地電腦的資料。
4. rsync 的角色就像是一般 Linux 的 cp 與 scp 指令，可以將檔案或目錄從來源位置複製到目的位置，不過 rsync 在複製檔案時會比 cp 與 scp 更有效率，並且支援連結檔與設備檔（devices），也可以保留檔案的擁有者、群組與權限設定。
5. rsync 在第一次複製檔案時，會複製完整的檔案內容，而之後再次複製檔案時，就會先以 delta transfer 演算法檢查新舊檔案之間的差異，只傳送有變動的部份，可加快備份速度，尤其是在累進備份大檔案時，效果更明顯。
6. rsync 在使用網路傳送資料時，也支援資料的自動壓縮與解壓縮，可以有效減少耗費的網路頻寬。

8.2 安裝 Rsync

1. 使用 yum install 安裝 rsync 工具。

```
[root@kvm5 ~]# yum install rsync
```

2. rsync 輔助說明

```
[root@kvm5 ~]# rsync --help | head
rsync  version 3.0.9  protocol version 30
Copyright (C) 1996-2011 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
    64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints,
    socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace,
```

```
    append, ACLs, xattrs, iconv, symtimes

rsync comes with ABSOLUTELY NO WARRANTY.  This is free software, and you
are welcome to redistribute it under certain conditions.  See the GNU
```

3. 可以 daemon 方式執行 rsync，輔助說明如下：

```
[root@kvm5 ~]# rsync --daemon --help
rsync  version 3.0.9  protocol version 30
Copyright (C) 1996-2011 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
    64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints,
    socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace,
    append, ACLs, xattrs, iconv, symtimes

rsync comes with ABSOLUTELY NO WARRANTY.  This is free software, and you
are welcome to redistribute it under certain conditions.  See the GNU
General Public Licence for details.

Usage: rsync --daemon [OPTION]...
     --address=ADDRESS       bind to the specified address
     --bwlimit=KBPS          limit I/O bandwidth; KBytes per second
     --config=FILE           specify alternate rsyncd.conf file
     --no-detach             do not detach from the parent
     --port=PORT             listen on alternate port number
     --log-file=FILE         override the "log file" setting
     --log-file-format=FMT   override the "log format" setting
     --sockopts=OPTIONS      specify custom TCP options
 -v, --verbose               increase verbosity
 -4, --ipv4                  prefer IPv4
 -6, --ipv6                  prefer IPv6
     --help                  show this help screen

If you were not trying to invoke rsync as a daemon, avoid using any of
    the
daemon-specific rsync options.  See also the rsyncd.conf(5) man page.
```

8.3 常用語法

1. -v：verbose 模式，輸出較詳細的訊息。
2. -r：遞迴（recursive），包含所有子目錄下的目錄與檔案。
3. -a：相當於 -rlptgoD，遞迴備份所有子目錄下的目錄與檔案，保留連結檔、檔案的擁有者、群組、權限以及時間戳記。
4. -z：啓用壓縮。

5. -h：將數字以比較容易閱讀的格式輸出。
6. --delete-excluded：刪除目標目錄中多的檔案。
7. --exclude：排除特定的檔案。
8. -u：跳過目標目錄中修改時間較來源端新的檔案。

8.4 基本用法一

1. 使用 -a,-v,-h 三個選項，將/etc/cron.daily 目錄備份到本機的/home/deyu1 目錄下。

```
[root@kvm5 ~]# rsync -avh /etc/cron.daily /home/deyu1/
sending incremental file list
cron.daily/
cron.daily/0yum-daily.cron
cron.daily/logrotate
cron.daily/man-db.cron

sent 1.38K bytes  received 73 bytes  2.91K bytes/sec
total size is 1.13K  speedup is 0.78
```

2. 使用/home/deyu1 目錄下，已有 cron.daily 目錄，且目錄下有三個檔案。

```
[root@kvm5 ~]# ll /home/deyu1/cron.daily/
total 6
-rwxr-xr-x. 1 root root 332 Mar 10  2015 0yum-daily.cron
-rwx------. 1 root root 180 Jul 31  2013 logrotate
-rwxr-xr-x. 1 root root 618 Mar 18  2014 man-db.cron
```

3. 再執行一次 rsync，因爲來源與目的端內容一樣，沒有任何檔案複製。

```
[root@kvm5 ~]# rsync -avh /etc/cron.daily /home/deyu1/
sending incremental file list

sent 121 bytes  received 13 bytes  268.00 bytes/sec
total size is 1.13K  speedup is 8.43
```

4. 刪除/home/deyu1/cron.daily/0yum-daily.cron。

```
[root@kvm5 ~]# rm /home/deyu1/cron.daily/0yum-daily.cron
rm: remove regular file ‘/home/deyu1/cron.daily/0yum-daily.’ cron? y
```

5. 再執行一次 rsync，因目的端缺 0yum-daily.cron，其他與來源端一樣，所以只複製此檔。

```
[root@kvm5 ~]# rsync -avh /etc/cron.daily /home/deyu1/
sending incremental file list
cron.daily/
cron.daily/0yum-daily.cron

sent 499 bytes  received 35 bytes  1.07K bytes/sec
total size is 1.13K  speedup is 2.12
```

8.5 基本用法二

1. 在/home/deyu1/cron.daily/logrotate 檔案最後加入一行'rsync test'。

```
[root@kvm5 ~]# echo 'rsync test' >> /home/deyu1/cron.daily/logrotate
```

2. 比較來源端與目的端的 logrotate，檔案內容不一樣。

```
[root@kvm5 ~]# diff -uN /etc/cron.daily/logrotate /home/deyu1/cron.daily
    /logrotate
--- /etc/cron.daily/logrotate   2013-07-31 19:46:23.000000000 +0800
+++ /home/deyu1/cron.daily/logrotate    2019-10-11 16:07:37.000000000
    +0800
@@ -6,3 +6,4 @@
     /usr/bin/logger -t logrotate "ALERT exited abnormally with [
         $EXITVALUE]"
 fi
 exit 0
+rsync test
```

3. 執行 rsync，只同步不一樣的檔案 logrotate。

```
[root@kvm5 ~]# rsync -avh /etc/cron.daily /home/deyu1/sending
    incremental file list
cron.daily/logrotate

sent 344 bytes  received 32 bytes  752.00 bytes/sec
total size is 1.13K  speedup is 3.01
```

8.6 用法三 -檔案新舊

1. 再一次變更，在/home/deyu1/cron.daily/logrotate 檔案最後加入一行'rsync test'。

```
[root@kvm5 ~]# echo 'rsync test' >> /home/deyu1/cron.daily/logrotate
```

2. 查看來源與目的端的 logrotate，目的端的檔比較新。

```
[root@kvm5 ~]# ll /etc/cron.daily/logrotate /home/deyu1/cron.daily/
    logrotate
-rwx------. 1 root root 180 Jul 31  2013 /etc/cron.daily/logrotate
-rwx------. 1 root root 191 Oct 11 16:15 /home/deyu1/cron.daily/
    logrotate
```

3. 執行 rsync 並增加 -u 選項，雖然來源與目的端的檔案 logrotate 不一樣，但目的端的檔案比較新，所以不會複製到目的端。

```
[root@kvm5 ~]# rsync -auvh /etc/cron.daily /home/deyu1/
sending incremental file list

sent 121 bytes  received 13 bytes  268.00 bytes/sec
total size is 1.13K  speedup is 8.43
```

8.7 用法四 -刪除排外

1. 在目端產生一個空檔 abc。

```
[root@kvm5 ~]# touch /home/deyu1/cron.daily/abc
```

2. 查看目的端目錄，多了一個檔案 abc。

```
[root@kvm5 ~]# ll /home/deyu1/cron.daily/
total 7
-rwxr-xr-x. 1 root root 332 Mar 10  2015 0yum-daily.cron
-rw-r--r--. 1 root root   0 Oct 11 16:27 abc
-rwx------. 1 root root 191 Oct 11 16:15 logrotate
-rwxr-xr-x. 1 root root 618 Mar 18  2014 man-db.cron
[root@kvm5 ~]# rsync -auvh /etc/cron.daily /home/deyu1/
sending incremental file list
cron.daily/

sent 124 bytes  received 16 bytes  280.00 bytes/sec
total size is 1.13K  speedup is 8.07
```

3. rsync 增加選項 --delete-excluded，同步後目的端多出來的檔案 abc 被刪除。

```
[root@kvm5 ~]# rsync -auvh --delete-excluded /etc/cron.daily /home/deyu1
    /
sending incremental file list
deleting cron.daily/abc

sent 121 bytes  received 13 bytes  268.00 bytes/sec
total size is 1.13K  speedup is 8.43
```

4. 查看目的端目錄檔案 abc 已被刪除。

```
[root@kvm5 ~]# ll /home/deyu1/cron.daily/
total 6
-rwxr-xr-x. 1 root root 332 Mar 10  2015 0yum-daily.cron
-rwx------. 1 root root 191 Oct 11 16:15 logrotate
-rwxr-xr-x. 1 root root 618 Mar 18  2014 man-db.cron
```

8.8 用法五 -排除特定檔案

1. 刪除之前同步的資料。

```
[root@kvm5 ~]# rm -rf /home/deyu1/cron.daily
```

2. 以選項 --exclude 排除副檔名是 cron 的檔案。

```
[root@kvm5 ~]# rsync -avh --exclude='*.cron' /etc/cron.daily /home/deyu1
    /
sending incremental file list
cron.daily/
cron.daily/logrotate

sent 293 bytes  received 35 bytes  656.00 bytes/sec
total size is 180  speedup is 0.55
```

3. 同步資料中只有 logrotate 一個檔。

```
[root@kvm5 ~]# ll /home/deyu1/cron.daily/
total 2
-rwx------. 1 root root 180 Jul 31  2013 logrotate
```

8.9 遠端同步

1. 將 deyu1 家目錄下的 cron.daily 目錄及目錄下的檔案，用户及群組變更爲 deyu1。

```
[root@kvm5 ~]# chown deyu1.deyu1 -R /home/deyu1/cron.daily
```

2. 切換身份成 deyu1。

```
[root@kvm5 ~]# su - deyu1
Last login: Fri Oct 11 15:46:34 CST 2019 on pts/0
```

3. 將 cron.daily 目錄同步到遠端機器 kvm7.deyu.wang (192.168.122.7) 帳號 deyu1 的家目錄，其中選項 -z 是將資料壓縮再傳送。

```
[deyu1@kvm5 ~]$ rsync -avzh cron.daily deyu1@kvm7.deyu.wang:
The authenticity of host 'kvm7.deyu.wang (192.168.122.7)' can't be
    established.
ECDSA key fingerprint is 3d:18:f7:73:34:1d:c1:fb:af:70:00:fa:e0:df:2d:ca
    .
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'kvm7.deyu.wang,192.168.122.7' (ECDSA) to the
     list of known hosts.
deyu1@kvm7.deyu.wang's password:
sending incremental file list
cron.daily/
cron.daily/0yum-daily.cron
cron.daily/logrotate
cron.daily/man-db.cron

sent 980 bytes  received 73 bytes  162.00 bytes/sec
total size is 1.14K  speedup is 1.08
```

4. 資料同步到 kvm7.deyu.wang 的 deyu1 家目錄了。

```
[root@kvm7 ~]# ll /home/deyu1/cron.daily/
total 6
-rwxr-xr-x. 1 deyu1 deyu1 332 Mar 10  2015 0yum-daily.cron
-rwx------. 1 deyu1 deyu1 191 Oct 11 16:15 logrotate
-rwxr-xr-x. 1 deyu1 deyu1 618 Mar 18  2014 man-db.cron
```

8.10 限制網路頻寬

1. 如果不希望 rsync 遠端備份資料時佔用太大的網路頻寬而影響正常的服務，可以加上 --bwlimit 選項來指定資料傳輸的速度上限，單位是 KBPS，以下是沒有限制頻寬的速度，每秒 4.34M。

```
[root@kvm5 ~]# rsync -avh /etc deyu1@kvm7.deyu.wang: | tail
deyu1@kvm7.deyu.wang's password:
etc/yum/pluginconf.d/langpacks.conf
etc/yum/pluginconf.d/priorities.conf
etc/yum/protected.d/
etc/yum/protected.d/systemd.conf
etc/yum/vars/
etc/yum/vars/contentdir
etc/yum/vars/infra

sent 19.51M bytes  received 19.08K bytes  4.34M bytes/sec
total size is 19.44M  speedup is 1.00
```

2. 刪除 kvm7.deyu.wang 上的同步資料。

```
[root@kvm7 ~]# rm -rf /home/deyu1/etc
```

3. 再以 bwlimit=100 參數，限制每秒 100K 的速度同步。

```
[root@kvm5 ~]# rsync -avh --bwlimit=100 /etc deyu1@kvm7.deyu.wang: |
    tail
deyu1@kvm7.deyu.wang's password:
etc/yum/pluginconf.d/langpacks.conf
etc/yum/pluginconf.d/priorities.conf
etc/yum/protected.d/
etc/yum/protected.d/systemd.conf
etc/yum/vars/
etc/yum/vars/contentdir
etc/yum/vars/infra

sent 19.51M bytes  received 19.08K bytes  100.39K bytes/sec
total size is 19.44M  speedup is 1.00
```

Chapter 9

LUKS 檔案系統加解密

9.1 前言

1. 參考資料
2. 硬碟加密可以防止他人取得硬碟後，讀取硬碟所有資料。
3. LUKS (Linux Unified Key Setup) 是一種區塊裝置 (block device) 加密規格。

9.2 LUKS 工具

1. 查看使用的硬碟編號爲/dev/sda，目前/ 根目錄掛載在/dev/sda3，預計將/dev/sda2 做 LUKS 加密並掛載成/ 根目錄。

```
[root@localhost ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0 18.4G  0 disk —
sda1   8:1    0  400M  0 part /boot —
sda2   8:2    0   10G  0 part —
sda3   8:3    0    4G  0 part / —
sda4   8:4    0    1K  0 part —
sda5   8:5    0  500M  0 part [SWAP]
```

2. 查詢提供 LUKS 工具命令的套件爲 cryptsetup。

```
[root@localhost ~]# yum provides *bin/cryptsetup
Loaded plugins: fastestmirror, langpacks
base                                                     | 3.6 kB
    00:00
extras                                                   | 3.4 kB
    00:00
updates                                                  | 3.4 kB
    00:00
(1/4): base/7/x86_64/group_gz                              | 156 kB
    00:00
(2/4): extras/7/x86_64/primary_db                          | 145 kB
    00:00
```

```
(3/4): updates/7/x86_64/primary_db                              | 4.6 MB
    00:05
(4/4): base/7/x86_64/primary_db                                 | 5.7 MB
    00:05
Determining fastest mirrors
 * base: ftp.twaren.net
 * extras: ftp.twaren.net
 * updates: ftp.twaren.net
base/7/x86_64/filelists_db                                    | 6.7 MB
    00:03
cryptsetup-1.7.4-3.el7.x86_64 : A utility for setting up encrypted disks
Repo        : base
Matched from:
Filename    : /usr/sbin/cryptsetup
```

3. 安裝 cryptsetup 套件。

```
[root@localhost ~]# yum install -y cryptsetup
```

4. 為加強分割區加密強度，格式化前最好先將整個分割區寫入隨機字元，以下為較快速但強度較低的寫入方式。

```
[root@localhost ~]# badblocks -c 10240 -s -w -t random -v /dev/sda2
Checking for bad blocks in read-write mode
From block 0 to 10485759
Testing with random pattern: done
Reading and comparing: done
Pass completed, 0 bad blocks found. (0/0/0 errors)
```

5. 使用 cryptsetup 對/dev/sda2 進行 luks 格式化，過程中先回答大寫的 YES，再輸入兩次密碼，密碼必須超過 7 個字元且不能為字典字。

```
[root@localhost ~]# cryptsetup luksFormat /dev/sda2

WARNING!
========
This will overwrite data on /dev/sda2 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Password quality check failed:
 The password is shorter than 7 characters
[root@localhost ~]# cryptsetup luksFormat /dev/sda2

WARNING!
========
This will overwrite data on /dev/sda2 irrevocably.
```

```
Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Password quality check failed:
 The password fails the dictionary check - it is based on a dictionary
    word
[root@localhost ~]# cryptsetup luksFormat /dev/sda2

WARNING!
========
This will overwrite data on /dev/sda2 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Password quality check failed:
 The password fails the dictionary check - it is based on a dictionary
    word
```

6. 成功進行/dev/sda2 分割區 LUKS 格式化。

```
[root@localhost ~]# cryptsetup luksFormat /dev/sda2

WARNING!
========
This will overwrite data on /dev/sda2 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase: ekC82B7XAmv3
Verify passphrase: ekC82B7XAmv3
```

7. 確認/dev/sda2 是否爲 LUKS 格式？沒有錯誤訊息表示正常。

```
[root@localhost ~]# cryptsetup isLuks /dev/sda2
[root@localhost ~]# cryptsetup isLuks /dev/sda2 && echo Success
Success
```

8. 使用 luksDump 查看/dev/sda2 的加密訊息。

```
[root@localhost ~]# cryptsetup luksDump /dev/sda2
LUKS header information for /dev/sda2

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha256
Payload offset: 4096
```

```
MK bits:        256
MK digest:      75 b9 1e 6c fc 72 6a c0 2a 67 6a db fa df 33 7f 65 a7 c5
    3b
MK salt:        d8 65 2c b2 f8 43 af f6 30 0b ca 49 01 bf 1c c7
                cd 62 7e 60 09 78 c8 5f 4f dc 5b f0 67 89 33 15
MK iterations:  89250
UUID:           b22f30aa-72d7-4523-aa21-dc516da53fc8

Key Slot 0: ENABLED
    Iterations:             703294
    Salt:                   50 b4 74 b0 1f 07 e0 e5 11 56 ae a7 2c bd 64
        a0
                            ef c6 e6 53 5c 73 5a 57 41 8f e4 38 69 99 2e
                                25
    Key material offset:    8
    AF stripes:             4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

9.3 LUKS Mapping

1. 要存取 LUKS 硬碟解密的內容，必須使用核心的 device-mapper 建立一個 mapping，建立前先查看 LUKS device 的 UUID。

```
[root@localhost ~]# cryptsetup luksUUID /dev/sda2
b22f30aa-72d7-4523-aa21-dc516da53fc8
```

2. 輸入 LUKS 格式化設定的密碼建立/dev/sda2 的 mapping，mapping 的名稱可自訂。

```
[root@localhost ~]# cryptsetup luksOpen /dev/sda2 luks-root
Enter passphrase for /dev/sda2:
```

3. 查看 luks-root 的訊息。

```
[root@localhost ~]# dmsetup info luks-root
Name:              luks-root
State:             ACTIVE
Read Ahead:        8192
Tables present:    LIVE
Open count:        0
```

```
Event number:       0
Major, minor:       253, 0
Number of targets: 1
UUID: CRYPT-LUKS1-b22f30aa72d74523aa21dc516da53fc8-luks-root
```

4. luks-root 已出現於/dev/mapper 目錄下。

```
[root@localhost ~]# ll /dev/mapper/luks-root
lrwxrwxrwx. 1 root root 7 Dec 28 11:29 /dev/mapper/luks-root -> ../dm-0
```

9.4 LUKS Mapping 掛載

1. 格式化 luks-root。

```
[root@localhost ~]# mkfs.xfs /dev/mapper/luks-root
meta-data=/dev/mapper/luks-root  isize=512    agcount=4, agsize=655232
    blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=2620928, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

2. 建立掛載目錄/sysroot。

```
[root@localhost ~]# mkdir /sysroot
```

3. 掛載/dev/mapper/luks-root 到/sysroot 目錄。

```
[root@localhost ~]# mount /dev/mapper/luks-root /sysroot/
```

4. 查看掛載。

```
[root@kvm7 ~]# df -h /sysroot
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/luks-root  2.0G  136M  1.8G   1% /sysroot
```

9.5 initramfs 處理

1. 預設的 Initialized RAM disk initramfs image 可能沒有 luks 的 driver，開機時無法開啟 luks 的 mapping，必須新增 dracut 參數，加入 lvm 及 crypt 模組，如果要用 USB 隨身碟放 LUKS 開機解密的 keyfile，也必須加入 `usb_storage` driver，ext4, vfat 等隨身碟分割區可能的格式，否則無法掛載。

```
[root@kvm7 ~]# vim /etc/dracut.conf.d/autounlock.conf
[root@kvm7 ~]# cat /etc/dracut.conf.d/autounlock.conf
# dracut modules to omit
omit_dracutmodules+=" systemd"

# dracut modules to add to the default
add_dracutmodules+=" lvm crypt"

# dracut drivers to add to the default
add_drivers+=" usb_storage xts aes-x86_64"

filesystems+=" ext4 vfat xfs"
```

2. dracut 產生 initramfs 檔。

```
[root@kvm7 ~]# dracut /boot/initramfs-`uname -r`-luks-root.img `uname -r
    `
```

3. 查看/boot 目錄下的 initramfs 檔，多了 initramfs-xxxx-luks-root.img。

```
[root@kvm7 ~]# ll /boot/initramfs-`uname -r`*
-rw-------. 1 root root 26460783 May 20  2019 /boot/initramfs
    -3.10.0-862.2.3.el7.x86_64.img
-rw-------. 1 root root 26821231 Dec  5 13:04 /boot/initramfs
    -3.10.0-862.2.3.el7.x86_64-luks-root.img
```

4. initrd 使用新建的 initramfs-xxxx-luks-root.img。

```
[root@kvm7 ~]# vim /boot/grub2/grub.cfg
[root@kvm7 ~]# grep luks /boot/grub2/grub.cfg
    initrd16 /initramfs-3.10.0-862.2.3.el7.x86_64-luks-root.img
```

9.6 LUKS 安全漏洞

1. linux 遠端利用漏洞 CVE-2016-4484。

2. Linux 磁碟加密工具 Cryptsetup 爆重大漏洞，駭客可取得系統根權限。
3. LUKS 密碼檢查允許使用者重覆輸入錯誤的密碼，若一直按著 Enter 鍵約 70 秒，系統會質疑是硬體故障，開始執行一個具有 root 權限的 initrd shell。
4. CVE-2016-4484 詳細描述與修復。

9.7 使用/etc/crypttab

1. 查詢/dev/vdb1 的 luksUUID。

```
[root@kvm7 ~]# cryptsetup luksUUID /dev/vdb1
77472cc9-0dd3-4c4c-81ad-e0ecae7570b2
```

2. 建立 luks 分割區對應檔，此檔有四個欄位，依序為 mapping 名、裝置、金鑰、格式，詳細的設定方式可用 man 5 crypttab 查詢設定手冊。

```
[root@kvm7 ~]# vim /etc/crypttab
[root@kvm7 ~]# cat /etc/crypttab
luks-root UUID=77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 none noauto
```

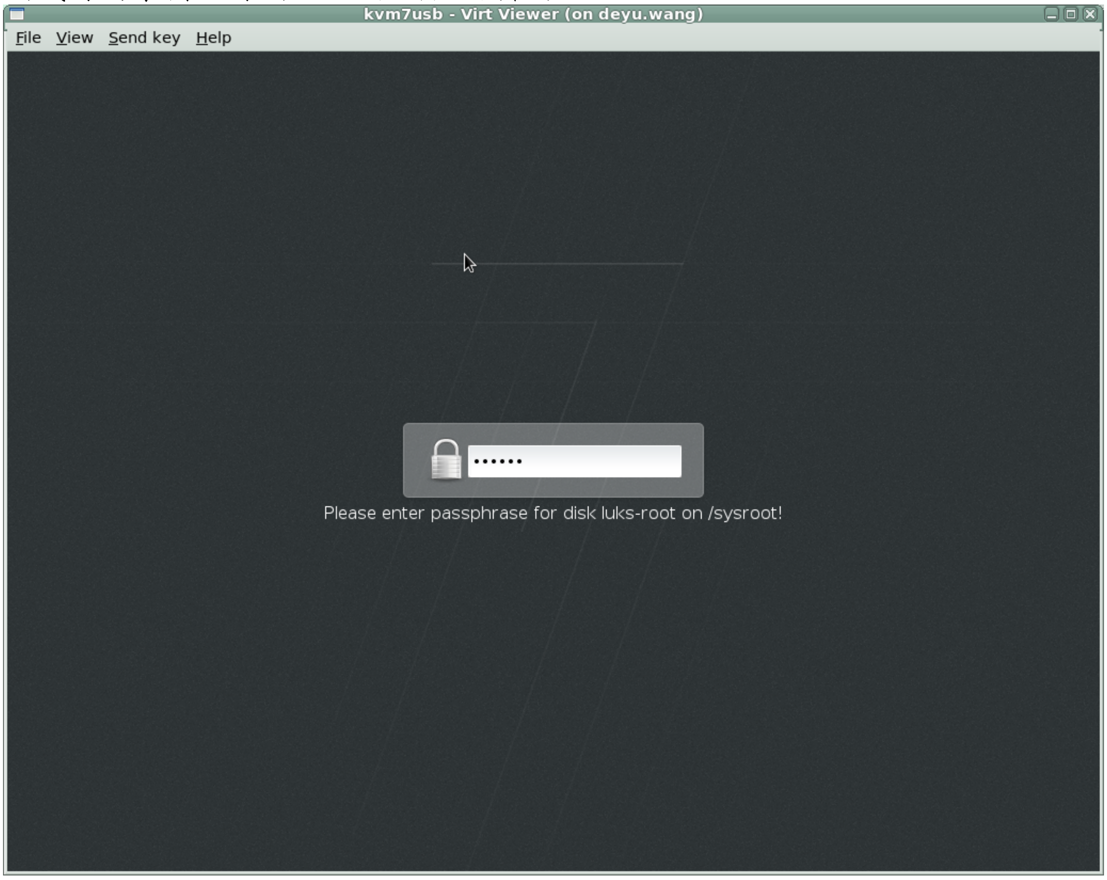
3. 編輯/etc/fstab，將/dev/mapper/luks-root 掛載於/sysroot 目錄，掛載參數 `_netdev` 將 luks 分割區視為外部硬碟，以免掛載不成功而 sulogin，參數 `x-systemd.device-timeout=1` 當 device 找不到時，1 秒後放棄掛載。

```
[root@kvm7 ~]# vim /etc/fstab
[root@kvm7 ~]# grep luks-root /etc/fstab
/dev/mapper/luks-root    /sysroot     xfs
 _netdev,x-systemd.device-timeout=1,defaults     0 0
```

4. 重新開機

```
[root@kvm7 ~]# reboot
```

5. 開機時要求輸入打開 luks 分割區的密碼。



6. 重新開機後，加密分割區/dev/vdb1 的 mapping luks-root，成功掛載在/sysroot 目錄。

```
[root@kvm7 ~]# lsblk /dev/vdb
NAME          MAJ:MIN RM SIZE RO TYPE  MOUNTPOINT
vdb           252:16   0   4G  0 disk —
vdb1         252:17   0   2G  0 part —
  luks-root 253:4    0   2G  0 crypt /sysroot
[root@kvm7 ~]# df -h /sysroot
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/luks-root  2.0G  1.7G  318M  85% /sysroot
```

9.8 KeyFile 解密

1. 產生一個隨機金鑰 keyfile。

```
[root@kvm7 ~]# dd if=/dev/urandom of=/boot/keyfile bs=1024 count=4
4+0 records in
4+0 records out
4096 bytes (4.1 kB) copied, 0.00599741 s, 683 kB/s
```

2. 將 keyfile 設定成只有 root 可讀取。

```
[root@kvm7 ~]# chmod 0400 /boot/keyfile
```

3. 輸入先前 luks 格式化設定的密碼將 keyfile 加到 luks 分割區/dev/vdb1。

```
[root@kvm7 ~]# cryptsetup luksAddKey /dev/vdb1 /boot/keyfile
Enter any existing passphrase:
```

4. 如果之前已開啓 luks 的 mapping，則先卸載/sysroot，並移除 mapping /dev/mapper/luks-root。

```
[root@kvm7 ~]# umount /sysroot/
[root@kvm7 ~]# dmsetup remove /dev/mapper/luks-root
[root@kvm7 ~]# ll /dev/mapper/luks-root
ls: cannot access /dev/mapper/luks-root: No such file or directory
```

5. 以/boot/keyfile 開啓/dev/vdb1 mapping。

```
[root@kvm7 ~]# cryptsetup --key-file /boot/keyfile luksOpen /dev/vdb1
    luks-root
[root@kvm7 ~]# ll /dev/mapper/luks-root
lrwxrwxrwx. 1 root root 7 Dec  5 14:14 /dev/mapper/luks-root -> ../dm-4
```

6. 修改 luks mapping 檔/etc/crypttab 分割區自動解密對應檔，此檔有四個欄位，依序爲 mapping 名、裝置、金鑰、格式。

```
[root@kvm7 ~]# vim /etc/crypttab
[root@kvm7 ~]# cat /etc/crypttab
luks-root UUID=77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 /boot/keyfile luks
```

7. 重新開機

```
[root@kvm7 ~]# reboot
```

8. 重開機不用輸入密碼，加密分割區/dev/vdb1 的 mapping luks-root，成功掛載在/sysroot 目錄。

```
[root@kvm7 ~]# lsblk /dev/vdb
NAME          MAJ:MIN RM SIZE RO TYPE  MOUNTPOINT
vdb           252:16   0   4G  0 disk
vdb1          252:17   0   2G  0 part
  luks-root 253:4    0   2G  0 crypt /sysroot
[root@kvm7 ~]# df -h /sysroot
Filesystem             Size  Used Avail Use% Mounted on
/dev/mapper/luks-root  2.0G  1.7G  318M  85% /sysroot
```

9.9 不使用/etc/crypttab

1. 沒有/etc/crypttab 檔做 luks mapping 名稱對應時，luks 解密的 mapping 爲 luks-uuid，查詢/dev/vdb1 的 uuid。

```
[root@kvm7 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="77472cc9-0dd3-4c4c-81ad-e0ecae7570b2" TYPE="crypto_LUKS"
```

2. 開機掛載表/etc/fstab 的 luks mapping 改成 luks-uuid 型式，掛載參數 x-systemd.device-timeout=1 當 device 找不到時，1 秒後放棄掛載。

```
[root@kvm7 ~]# grep luks /etc/fstab
/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2   /sysroot    xfs
 _netdev,x-systemd.device-timeout=1,defaults      0 0
```

3. 開機核心參數指定 rd.luks.uuid 及 rd.luks.key，金鑰的格式是 <key-path>:<keydev> keydev 可用 UUID 或 LABEL，rd.luks.crypttab=no 指定不使用/etc/crypttab 做對應，否則開機時依然會搜尋 device，直到 timeout。

```
[root@kvm7 ~]# grep luks /boot/grub2/grub.cfg
    linux16 /vmlinuz-3.10.0-862.2.3.el7.x86_64 root=/dev/mapper/
        vg_kvm7usb-root ro
 crashkernel=auto rd.lvm.lv=vg_kvm7usb/root rd.lvm.lv=vg_kvm7usb/swap
    rhgb quiet
 rd.luks.uuid=77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 rd.luks.name=luks-
    root
 rd.luks.key=/keyfile:UUID="ad20c60f-2418-488b-950f-90fc56cc7792" rd.
    luks.crypttab=no
    initrd16 /initramfs-3.10.0-862.2.3.el7.x86_64-luks-root.img
```

4. 重新開機

```
[root@kvm7 ~]# reboot
```

5. 重開機不用輸入密碼，加密分割區/dev/vdb1 的 mapping luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2，成功掛載在/sysroot 目錄。

```
[root@kvm7 ~]# lsblk /dev/vdb
NAME                                           MAJ:MIN RM SIZE RO TYPE
    MOUNTPOINT
vdb                                            252:16   0   4G  0 disk —
vdb1                                         252:17   0   2G  0 part —
  luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2
                                               253:0    0   2G  0 crypt /
    sysroot
[root@kvm7 ~]# df -h /sysroot/
Filesystem                                            Size  Used Avail
    Use% Mounted on
/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2  2.0G  1.7G  318M
     85% /sysroot
```

Chapter 10

LUKS 加密根目錄

10.1 根目錄檔案複製

1. 格式化/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2。

```
[root@kvm7 ~]# mkfs.xfs /dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 -f
meta-data=/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 isize=512    agcount=4, agsize=130944 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=523776, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

2. 建立掛載目錄/sysroot 掛載/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2。

```
[root@kvm7 ~]# mkdir /sysroot
[root@kvm7 ~]# mount /dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 /sysroot/
[root@kvm7 ~]# ll /sysroot
total 0
```

3. 將/根目錄檔案系統複製到 luks 掛載目錄/sysroot。

```
[root@kvm7 ~]# cp -axZ / /sysroot/
[root@kvm7 ~]# ll /sysroot/
total 20
lrwxrwxrwx.  1 root root    7 Jul 10  2018 bin -> usr/bin
dr-xr-xr-x.  2 root root    6 May 20  2019 boot
drwxr-xr-x.  2 root root    6 Dec  4 19:51 dev
drwxr-xr-x. 89 root root 8192 Dec  4 21:29 etc
```

```
drwxr-xr-x.  2 root root    6 Jun 10 21:09 home
lrwxrwxrwx.  1 root root    7 Jul 10  2018 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 10  2018 lib64 -> usr/lib64
drwx------.  2 root root    6 Jul 10  2018 lost+found
drwxr-xr-x.  2 root root    6 Apr 11  2018 media
drwxr-xr-x.  2 root root    6 Dec  4 16:33 misc
drwxr-xr-x.  5 root root   46 Oct 24 19:51 mnt
drwxr-xr-x.  2 root root    6 Dec  4 16:33 net
drwxr-xr-x.  2 root root    6 Apr 11  2018 opt
dr-xr-xr-x.  2 root root    6 Sep  1 17:20 proc
dr-xr-x---.  5 root root 4096 Dec  4 19:23 root
drwxr-xr-x.  2 root root    6 Dec  4 19:19 run
lrwxrwxrwx.  1 root root    8 Jul 10  2018 sbin -> usr/sbin
drwxr-xr-x.  2 root root    6 Apr 11  2018 srv
dr-xr-xr-x.  2 root root    6 Sep  1 17:20 sys
drwxr-xr-x.  2 root root    6 Dec  4 20:04 sysroot
drwxrwxrwt. 12 root root 4096 Dec  4 21:43 tmp
drwxr-xr-x. 13 root root  155 Jul 10  2018 usr
drwxr-xr-x. 20 root root  278 Jul 10  2018 var
```

4. 要記得在/sysroot 的根目錄產生一個空檔.autorelabel，在以 luks mapping 爲根目錄開機時，恢復檔案的 selinux contexts，否則開機時會因 selinux 在啓動狀態而無法登入。

```
[root@kvm7 ~]# touch /sysroot/.autorelabel
```

5. 修改 luks 掛載目錄/sysroot 的開機掛載表 etc/fstab，根目錄掛載裝置改爲/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2。

```
[root@kvm7 ~]# grep luks /sysroot/etc/fstab
/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2    /     xfs
    defaults    1 1
```

10.2 根目錄 LUKS 加密

1. 開機選單新增一根目錄爲 luks-uuid 的檔案系統，核心參數指定 rd.luks.uuid 及 rd.luks.key，金鑰的格式是 <keypath>:<keydev> keydev 可用 UUID 或 LABEL，rd.luks.crypttab=no 指定不使用/etc/crypttab 檔的 luks mapping 對應解密檔。

```
[root@kvm7 ~]# vim /boot/grub2/grub.cfg
[root@kvm7 ~]# grep luks /boot/grub2/grub.cfg
    linux16 /vmlinuz-3.10.0-862.2.3.el7.x86_64
 root=/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 ro
     crashkernel=auto
 rd.lvm.lv=vg_kvm7usb/swap rhgb quiet rd.luks.uuid=77472cc9-0dd3-4c4c-81
     ad-e0ecae7570b2
```

```
rd.luks.name=luks-root rd.luks.key=/keyfile:UUID="ad20c60f-2418-488b
    -950f-90fc56cc7792"
rd.luks.crypttab=no
   initrd16 /initramfs-3.10.0-862.2.3.el7.x86_64-luks-root.img

   linux16 /vmlinuz-3.10.0-862.2.3.el7.x86_64 root=/dev/mapper/
       vg_kvm7usb-root ro
crashkernel=auto rd.lvm.lv=vg_kvm7usb/root rd.lvm.lv=vg_kvm7usb/swap
    rhgb quiet
   initrd16 /initramfs-3.10.0-862.2.3.el7.x86_64.img
```

2. 重新開機

```
[root@kvm7 ~]# reboot
```

3. 重開機以 luks 選單開機，不用輸入密碼，開機後登入查看根目錄掛載在/dev/dm-0。加密分割區/dev/vdb1 的 mapping luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2，成功掛載在/ 根目錄。

```
[root@kvm7 ~]# df -h
Filesystem                    Size  Used Avail Use% Mounted on
/dev/dm-0                     2.0G  1.6G  440M  79% /
devtmpfs                      483M     0  483M   0% /dev
tmpfs                         496M     0  496M   0% /dev/shm
tmpfs                         496M  6.8M  490M   2% /run
tmpfs                         496M     0  496M   0% /sys/fs/cgroup
/dev/mapper/vg_kvm7home-vo     35M   33M  193K 100% /home
/dev/vda1                     197M  160M   38M  82% /boot
tmpfs                         100M     0  100M   0% /run/user/0
```

4. 查看/dev/dm-0 就是 luks 分割區/dev/vdb1 的 mapping luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2。

```
[root@kvm7 ~]# ll /dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2
lrwxrwxrwx. 1 root root 7 Dec  5 20:56 /dev/mapper/luks-77472cc9-0dd3-4
    c4c-81ad-e0ecae7570b2 -> ../dm-0
[root@kvm7 ~]# lsblk /dev/vdb
NAME                                            MAJ:MIN RM SIZE RO TYPE
    MOUNTPOINT
vdb                                             252:16   0   4G  0 disk ─
vdb1                                          252:17   0   2G  0 part ─
  luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2
                                                253:0    0   2G  0 crypt /
```

10.3 keyfile 存放檔案系統準備

1. 本節之前爲方便測試，LUKS 的解密檔放在開機檔案系統/boot 中，實際使用時必須將 keyfile 存放於其他裝置，例如：USB 隨身碟，此時如果使用 UUID 指定裝置，就只能固定使一個 usb 碟開機，所以必須改用裝置 label。查看目前的檔案系統，利用/dev/vda6 的 vgmnt 來測試不同格式的檔案系統存放 luks 金鑰。

```
[root@kvm7 ~]# lsblk
NAME                                          MAJ:MIN RM  SIZE RO TYPE
    MOUNTPOINT
vda                                           252:0    0    4G  0 disk
vda1                                        252:1    0  200M  0 part  /
    boot
vda2                                        252:2    0  3.4G  0 part
 vg_kvm7usb-root                          253:1    0  3.1G  0 lvm   /
     sysroot
 vg_kvm7usb-swap                          253:2    0  124M  0 lvm   [SWAP
     ]
vda3                                        252:3    0  130M  0 part
 vg_kvm7home-vo                           253:4    0   40M  0 lvm   /home
vda4                                        252:4    0    1K  0 part
vda5                                        252:5    0   15M  0 part  [
    SWAP]
vda6                                        252:6    0  248M  0 part
  vgmnt-ppp                                 253:3    0   16M  0 lvm
vdb                                           252:16   0    4G  0 disk
vdb1                                        252:17   0    2G  0 part
  luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2
                                              253:0    0    2G  0 crypt /
```

2. 修改 vgmnt 卷冊組中的 lv ppp 名稱爲 xfs，而且再產生名爲 ext4 及 vfat 兩個 lv。

```
[root@kvm7 ~]# lvrename vgmnt ppp xfs
  Renamed "ppp" to "xfs" in volume group "vgmnt"
[root@kvm7 ~]# lvcreate -L 10M -n ext4 vgmnt
  Rounding up size to full physical extent 12.00 MiB
  Logical volume "ext4" created.
[root@kvm7 ~]# lvcreate -L 10M -n vfat vgmnt
  Rounding up size to full physical extent 12.00 MiB
  Logical volume "vfat" created.
```

3. vgmnt 卷冊組中的名爲 xfs, ext4 及 vfat 三個 lv，分別格式化爲 xfs, ext4 及 vfat。

```
[root@kvm7 ~]# mkfs.xfs /dev/vgmnt/xfs
mkfs.xfs: /dev/vgmnt/xfs appears to contain an existing filesystem (ext3
    ).
mkfs.xfs: Use the -f option to force overwrite.
[root@kvm7 ~]# mkfs.xfs /dev/vgmnt/xfs -f
```

```
meta-data=/dev/vgmnt/xfs         isize=512    agcount=1, agsize=4096
    blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=4096, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=855, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@kvm7 ~]# mkfs.ext4 /dev/vgmnt/ext4
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
3072 inodes, 12288 blocks
614 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=12582912
2 block groups
8192 blocks per group, 8192 fragments per group
1536 inodes per group
Superblock backups stored on blocks:
    8193

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

[root@kvm7 ~]# mkfs.vfat /dev/vgmnt/vfat
mkfs.fat 3.0.20 (12 Jun 2013)
unable to get drive geometry, using default 255/63
```

4. 建立並掛載 xfs, ext4 及 vfat 三個 lv。

```
[root@kvm7 ~]# mkdir -p /mnt/lukskey/{xfs,ext4,vfat}
[root@kvm7 ~]# mount /dev/vgmnt/xfs /mnt/lukskey/xfs
[root@kvm7 ~]# mount /dev/vgmnt/ext4 /mnt/lukskey/ext4
[root@kvm7 ~]# mount /dev/vgmnt/vfat /mnt/lukskey/vfat
```

10.4 新增 LUKS 的 keyfiles

1. 在 xfs, ext4 及 vfat 三個 lv 掛載目錄下，各新增一把金鑰。

```
[root@kvm7 ~]# dd if=/dev/urandom of=/mnt/lukskey/xfs/xfskeyfile bs=1024
    count=4
```

```
4+0 records in
4+0 records out
4096 bytes (4.1 kB) copied, 0.000942181 s, 4.3 MB/s
[root@kvm7 ~]# dd if=/dev/urandom of=/mnt/lukskey/ext4/ext4keyfile bs
    =1024 count=4
4+0 records in
4+0 records out
4096 bytes (4.1 kB) copied, 0.000761101 s, 5.4 MB/s
[root@kvm7 ~]# dd if=/dev/urandom of=/mnt/lukskey/vfat/vfatkeyfile bs
    =1024 count=4
4+0 records in
4+0 records out
4096 bytes (4.1 kB) copied, 0.00172373 s, 2.4 MB/s
```

2. 將 xfs, ext4 及 vfat 三個 lv 掛載目錄下的金鑰，都新增爲開啓/dev/vdb1 的金鑰。

```
[root@kvm7 ~]# cryptsetup luksAddKey /dev/vdb1 /mnt/lukskey/xfs/
    xfskeyfile
Enter any existing passphrase:
[root@kvm7 ~]# cryptsetup luksAddKey /dev/vdb1 /mnt/lukskey/ext4/
    ext4keyfile
Enter any existing passphrase:
[root@kvm7 ~]# cryptsetup luksAddKey /dev/vdb1 /mnt/lukskey/vfat/
    vfatkeyfile
Enter any existing passphrase:
```

3. 查看/dev/vdb1 的金鑰，目前有 Key Slot 0 至 4 共 5 把金鑰。

```
[root@kvm7 ~]# cryptsetup luksDump /dev/vdb1
LUKS header information for /dev/vdb1

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha256
Payload offset: 4096
MK bits:        256
MK digest:      dc f8 00 ba 18 9d ee 80 80 27 c9 5e 2b e4 88 5e e0 05 d6
    39
MK salt:        d1 3e 91 cb 80 f3 4f 9d 91 c4 da e2 25 e2 5b 5e
                78 d5 07 de 04 6e fd 9e 48 4d 8a 31 49 42 16 f5
MK iterations:  35008
UUID:           77472cc9-0dd3-4c4c-81ad-e0ecae7570b2

Key Slot 0: ENABLED
    Iterations:             556568
    Salt:                   6f ea 6a 2c c8 4b 97 d2 47 c9 6b c8 ef b2 d7
         86
                            a6 58 41 14 8f 34 3f a9 d4 25 02 6c 0e c8 a6
                                 12
    Key material offset:    8
```

```
    AF stripes:               4000
Key Slot 1: ENABLED
    Iterations:               563144
    Salt:                     66 0e 2e c1 8a a3 3f cb 21 7d 50 ed aa 96 3d
         d7
                              09 54 36 a9 15 5a b1 3e 4e fe 35 d6 10 b3 ac
                                   35
    Key material offset:      264
    AF stripes:               4000
Key Slot 2: ENABLED
    Iterations:               517558
    Salt:                     59 ef 20 e6 1b da 6e b9 03 0f c4 8c 4d 1a 57
         71
                              5d 18 92 e8 da 42 b9 9b 00 0f f2 17 ee e1 10
                                   41
    Key material offset:      520
    AF stripes:               4000
Key Slot 3: ENABLED
    Iterations:               539946
    Salt:                     8b 0e d0 09 ad 19 77 51 a6 5c 61 24 37 24 a6
         02
                              f2 95 02 6e 36 52 52 fb 2a 24 66 56 ad 54 2f
                                   fd
    Key material offset:      776
    AF stripes:               4000
Key Slot 4: ENABLED
    Iterations:               520642
    Salt:                     7f 34 1c 22 4d 97 1d 9f d2 fe fb 8b 4c 47 42
         de
                              b0 a5 6d 82 54 0e 26 65 f5 4f ac 9e 9e 8a 01
                                   33
    Key material offset:      1032
    AF stripes:               4000
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

10.5 檔案系統設定 Label

1. 查看三個存放金鑰的 LV 都沒有設定 label。

```
[root@kvm7 ~]# blkid /dev/vgmnt/*
/dev/vgmnt/ext4: UUID="f6827c85-3054-46b6-8552-e18da8a7dfd8" TYPE="ext4"
/dev/vgmnt/vfat: SEC_TYPE="msdos" UUID="76C0-ED83" TYPE="vfat"
/dev/vgmnt/xfs: UUID="a6b24877-cb4e-49b5-ade1-ee19fdc2941e" TYPE="xfs"
```

2. 先卸載。

```
[root@kvm7 ~]# umount /mnt/lukskey/*
```

3. 設定 xfs 檔案系統的 label。

```
[root@kvm7 ~]# xfs_admin -L xfs-lukskey /dev/vgmnt/xfs
writing all SBs
new label = "xfs-lukskey"
```

4. 設定 ext 檔案系統的 label。

```
[root@kvm7 ~]# e2label /dev/vgmnt/ext4 ext4-lukskey
```

5. 設定 vfat 檔案系統的 label，長度有限制而且要大寫。

```
[root@kvm7 ~]# fatlabel /dev/vgmnt/vfat vfat-lukskey
fatlabel: labels can be no longer than 11 characters
[root@kvm7 ~]# fatlabel /dev/vgmnt/vfat vfat-luks
fatlabel: labels cannot contain lower case characters
[root@kvm7 ~]# fatlabel /dev/vgmnt/vfat VFAT-LUKS
```

6. 再查看三個存放金鑰的 LV 的 label。

```
[root@kvm7 ~]# blkid /dev/vgmnt/*
/dev/vgmnt/ext4: LABEL="ext4-lukskey" UUID="f6827c85-3054-46b6-8552-
    e18da8a7dfd8" TYPE="ext4"
/dev/vgmnt/vfat: SEC_TYPE="msdos" LABEL="VFAT-LUKS" UUID="76C0-ED83"
    TYPE="vfat"
/dev/vgmnt/xfs: LABEL="xfs-lukskey" UUID="a6b24877-cb4e-49b5-ade1-
    ee19fdc2941e" TYPE="xfs"
```

10.6 開機測試

1. 修改開機核心選項 rd.luks.key 的裝置為 LABEL="VFAT-LUKS" 下的 vfatkeyfile，由於測試的檔案系統是 lvm，所以必須加 rd.lvm.lv=vgmnt/vfat。

```
[root@kvm7 ~]# vim /boot/grub2/grub.cfg
[root@kvm7 ~]# grep LABEL /boot/grub2/grub.cfg
    linux16 /vmlinuz-3.10.0-862.2.3.el7.x86_64
 root=/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 ro
 crashkernel=auto rd.lvm.lv=vg_kvm7usb/root rd.lvm.lv=vg_kvm7usb/swap
 rhgb quiet rd.lvm.lv=vgmnt/vfat rd.luks.uuid=77472cc9-0dd3-4c4c-81ad-
     e0ecae7570b2
```

```
rd.luks.name=luks-root rd.luks.key=/vfatkeyfile:LABEL="VFAT-LUKS" rd.
    luks.crypttab=no
```

2. 重新開機

```
[root@kvm7 ~]# reboot
```

3. 開機時找不到金鑰，輸入密碼開機，登入後掛載/dev/vgmnt/vfat 查看，金鑰的檔名並不是當時產生的 vfatkeyfile 而是 VFATKE-1，原因是 vfat 檔案系統檔案長度限制並用大寫。

```
[root@kvm7 ~]# mount /dev/vgmnt/vfat /mnt/lukskey/vfat
[root@kvm7 ~]# ll /mnt/lukskey/vfat
total 4
-rwxr-xr-x. 1 root root 4096 Dec  6 08:01 VFATKE~1
```

4. 測試結果，使用大寫的檔名 VFATKE-1，開機無法取得金鑰，檔名改爲小寫的 vkey。

```
[root@kvm7 ~]# mv /mnt/lukskey/vfat/VFATKE~1 /mnt/lukskey/vfat/vkey
```

5. 修改開機核心選項 rd.luks.key 的裝置爲 LABEL="VFAT-LUKS" 下的 vkey。

```
[root@kvm7 ~]# vim /boot/grub2/grub.cfg
[root@kvm7 ~]# grep LABEL /boot/grub2/grub.cfg
    linux16 /vmlinuz-3.10.0-862.2.3.el7.x86_64
 root=/dev/mapper/luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 ro
 crashkernel=auto rd.lvm.lv=vg_kvm7usb/root rd.lvm.lv=vg_kvm7usb/swap
 rhgb quiet rd.lvm.lv=vgmnt/vfat rd.luks.uuid=77472cc9-0dd3-4c4c-81ad-
    e0ecae7570b2
 rd.luks.name=luks-root rd.luks.key=/vkey:LABEL="VFAT-LUKS" rd.luks.
    crypttab=no
```

6. 再重新開機

```
```

7. 根目錄成功掛載在/dev/vdb1 的 mapping luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2。

```
[root@kvm7 ~]# lsblk /dev/vdb1
NAME                                        MAJ:MIN RM SIZE RO TYPE
    MOUNTPOINT
vdb1                                        252:17   0   2G  0 part —
luks-77472cc9-0dd3-4c4c-81ad-e0ecae7570b2 253:3    0   2G  0 crypt /
```

8. xfs 及 ext4 檔案系統，都不需修改金鑰檔名，成功開機掛載。

10.7 加大 LUKS 分割區

1. 查看目前分割區，根目錄使用/dev/sda2 LUKS 加密，要將/dev/sda3 併入 LUKS k 加密分割區。

```
[root@localhost ~]# lsblk
NAME                                   MAJ:MIN RM  SIZE RO TYPE
    MOUNTPOINT
sda                                      8:0    0 18.4G  0 disk —
sda1                                   8:1    0  400M  0 part  /boot —
sda2                                   8:2    0   10G  0 part  —
 luks-b22f30aa-72d7-4523-aa21-dc516da53fc8
                                       253:0    0   10G  0 crypt / —
sda3                                   8:3    0    4G  0 part —
sda4                                   8:4    0    1K  0 part —
sda5                                   8:5    0  500M  0 part  [SWAP] —
sda6                                   8:6    0   10M  0 part
sr0                                     11:0    1 55.3M  0 rom
```

2. GUEST fdisk 刪除/dev/sda2 及/dev/sda3，重新分割的/dev/sda2，其開始及結束 sector 包含原來的/dev/sda2 及/dev/sda3，也就是將/dev/sda2 及/dev/sda3 合併爲/dev/sda2。

```
[root@localhost ~]# fdisk /dev/sda
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p

Disk /dev/sda: 19.8 GB, 19787972608 bytes, 38648384 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000a6d01

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      821247      409600   83  Linux
```

```
/dev/sda2          821248   21792767   10485760   83  Linux
/dev/sda3        21792768   30181375    4194304   83  Linux
/dev/sda4        30181376   38647807    4233216    5  Extended
/dev/sda5        30183424   31207423     512000   82  Linux swap /
    Solaris
/dev/sda6        31209472   31229951      10240   83  Linux

Command (m for help): d
Partition number (1-6, default 6): 3
Partition 3 is deleted

Command (m for help): d
Partition number (1,2,4-6, default 6): 2
Partition 2 is deleted

Command (m for help): n
Partition type:
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p): p
Partition number (2,3, default 2): 2
First sector (821248-38648383, default 821248):
Using default value 821248
Last sector, +sectors or +size{K,M,G} (821248-30181375, default
    30181375):
Using default value 30181375
Partition 2 of type Linux and of size 14 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
    resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

3. GUEST 重新開機。

```
[root@localhost ~]# reboot
Connection to 127.0.0.1 closed by remote host.
Connection to 127.0.0.1 closed.
```

4. 重新登入 GUEST 後查看，LUKS 分割區已加大到 14G。

```
[root@dyw219 ~]# ssh -o VerifyHostKeyDNS=no -l kkdd -p 5679 127.0.0.1
Verification code:
Password:
Last login: Tue Mar 20 10:10:03 2018 from gateway
```

```
[kkdd@localhost ~]$ su -
Password:
Last login: Tue Mar 20 10:10:13 CST 2018 on pts/0
[root@localhost ~]# lsblk
NAME                                       MAJ:MIN RM  SIZE RO TYPE
    MOUNTPOINT
sda                                          8:0    0 18.4G  0 disk —
sda1                                       8:1    0  400M  0 part  /boot —
sda2                                       8:2    0   14G  0 part  —
 luks-b22f30aa-72d7-4523-aa21-dc516da53fc8
                                           253:0    0   14G  0 crypt / —
sda4                                       8:4    0    1K  0 part —
sda5                                       8:5    0  500M  0 part  [SWAP] —
sda6                                       8:6    0   10M  0 part
sr0                                         11:0    1 55.3M  0 rom
```

5. 但掛載在根目錄的分割區還是只有原來的 10G。

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/dm-0        10G  7.2G  2.8G  73% /
devtmpfs        2.1G     0  2.1G   0% /dev
tmpfs           2.1G     0  2.1G   0% /dev/shm
tmpfs           2.1G  8.9M  2.1G   1% /run
tmpfs           2.1G     0  2.1G   0% /sys/fs/cgroup
/dev/sda1       397M  248M  150M  63% /boot
tmpfs           422M   12K  422M   1% /run/user/42
tmpfs           422M     0  422M   0% /run/user/1000
tmpfs           422M     0  422M   0% /run/user/0
```

6. 必須先使用 cryptsetup resize luks 檔案系統。

```
[root@localhost ~]# cryptsetup -v resize luks-b22f30aa-72d7-4523-aa21-
    dc516da53fc8
Command successful.
```

7. 再使用 resize2fs 重新調整 luks 檔案系統，出現以下錯誤，原因爲檔案系統爲 xfs 格式，不是 ext2/ext3/ext4 格式，不能使用 resize2fs 工具。

```
[root@localhost ~]# resize2fs /dev/mapper/luks-b22f30aa-72d7-4523-aa21-
    dc516da53fc8
resize2fs 1.42.9 (28-Dec-2013)
resize2fs: Bad magic number in super-block while trying to open /dev/
    mapper/luks-b22f30aa-72d7-4523-aa21-dc516da53fc8
Couldn't find valid filesystem superblock.
```

8. 此 luks 分割區爲 xfs 檔案系統，要使用 xfs_growfs 工具將檔案系統放大到 cryptsetup resize 調整的值。

```
[root@localhost ~]# xfs_growfs /dev/mapper/luks-b22f30aa-72d7-4523-aa21-
    dc516da53fc8
meta-data=/dev/mapper/luks-b22f30aa-72d7-4523-aa21-dc516da53fc8 isize
    =512    agcount=4, agsize=655232 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=2620928, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 2620928 to 3669504
```

9. 再查看根目錄掛載點已放大到 14G。

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/dm-0        14G  7.2G  6.8G  52% /
devtmpfs        2.1G     0  2.1G   0% /dev
tmpfs           2.1G     0  2.1G   0% /dev/shm
tmpfs           2.1G  8.9M  2.1G   1% /run
tmpfs           2.1G     0  2.1G   0% /sys/fs/cgroup
/dev/sda1       397M  248M  150M  63% /boot
tmpfs           422M   12K  422M   1% /run/user/42
tmpfs           422M     0  422M   0% /run/user/1000
tmpfs           422M     0  422M   0% /run/user/0
```