

# Knowledgebase

## OpenBSD Kernel Module

*author: Jim*

Untuk dapat *meload* dan *unload kernel module* ketika proses development, Anda dapat mengubah *value securelevel* pada *kernel state* dari *1* menjadi *-1*. Modifikasi *value* tersebut dapat dilakukan pada file */etc/rc.securelevel*. Setelah itu, Anda dapat *meload* dan *unload kernel modules* tanpa perlu *reboot(8)*.

## Administrasi Remote System dengan SSH

*author: Chiank*

Kalau Anda mengelola puluhan *workstation* atau *server* berbasis UNIX, tentu sangat melelahkan ketika Anda harus login dan mengetikkan *password* ke setiap *remote system*. Ada beberapa cara untuk otomatisasi login tanpa password, seperti *rlogin*, menggunakan script, dsb. Tips kali ini akan menjelaskan otomatisasi login tanpa password dengan menggunakan fitur *authentication key* dari *ssh*.

### Apa yang Anda butuhkan?

- *remote system*: Anda membutuhkan service *ssh* dalam keadaan aktif. Anda dapat menggunakan *ssh* server dari SSH (SSH.COM) atau OpenSSH.
- *administration system*: Anda membutuhkan *ssh* client.

### Apa yang harus Anda lakukan?

- membuat *public key*
- mendistribusikan *public key* ke *remote system*

### Membuat public key

contoh yang saya lakukan :

```
$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/chiank/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/chiank/.ssh/id_dsa.
Your public key has been saved in /home/chiank/.ssh/id_dsa.pub.
```

Pada contoh diatas, kita membuat file *secret key id\_dsa* dan *public key id\_dsa.pub*. Karena tipe yang kita gunakan adalah *dsa* (lihat *option -t dsa*, *key* yang kita miliki bisa digunakan untuk *ssh* versi 2,

sedangkan untuk ssh versi 1, gunakan option `-t rsa`.

Ketika mendapatkan *prompt passphrase* Anda dapat mengosongkannya dengan menekan tombol `ENTER`, atau Anda membuat password tersendiri.

**Perlu diingat:** *passphrase* ini berbeda dengan password yang Anda gunakan pada *remote system* tersebut.

## Mendistribusikan public key ke remote system

Langkah selanjutnya adalah memasukkan entry *public key* ke file `$HOME/.ssh/authorized_keys` (untuk beberapa system dan versi ssh, file yang digunakan adalah `$HOME/.ssh/authorized_keys2`). Langkah ini dapat dilakukan dengan berbagai cara, salah satunya dengan menggunakan `scp` untuk mengirimkan *public key* yang sudah dibuat tersebut ke *remote system* dan memasukkannya kedalam file `$HOME/.ssh/authorized_keys` atau dengan melakukan *copy & paste* menggunakan terminal (`xterm` atau `putty`)

```
$ scp ~/.ssh/id_dsa.pub chiank@tb.ee.itb.ac.id:
chiank@tb.ee.itb.ac.id's password:
id_dsa.pub                                100% 604      0.0KB/s
00:00
$ ssh chiank@tb.ee.itb.ac.id
chiank@tb.ee.itb.ac.id's password:
...
tb.ee $ cat id_dsa.pub >> $HOME/.ssh/authorized_keys
tb.ee $ rm -f id_dsa.pub
tb.ee $ exit
...
$ ssh chiank@tb.ee.itb.ac.id
tb.ee $
```

## Bikin file kosong dengan ukuran tertentu

*author: Jim*

[dari milis id-freebsd@yahoogroups.com](mailto:id-freebsd@yahoogroups.com)

```
> gimana caranya bikin file kosong
> dng ukuran tertentu (mis 1 MB, 2 MB, 4 MB, dst nya)
```

```
$ mkfile nnn[k|b|m] filename
=(/usr/ports/sysutils/mkfile)
```

```
develbox:/tmp > mkfile 10m file10M
develbox:/tmp > du -sh file10M
10M file10M
develbox:/tmp > strings file10M
develbox:/tmp >
```

kalau mau bikin banyak file sekaligus.. (pake tcsh 😊)

```
develbox:/tmp > foreach i ( `jot 5` )
foreach? mkfile "$i"m file"$i"m
foreach? end
develbox:/tmp > du -sh file*
1.0M file1m
2.0M file2m
3.0M file3m
4.0M file4m
5.0M file5m
```

## Bikin file ISO OpenBSD

Ini cara singkat untuk membuat file ISO OpenBSD untuk 3.8-RELEASE/i386.

### Persiapkan direktori kerja

Anggap kita akan menggunakan direktori \$HOME sebagai direktori kerja

```
$ mkdir -p $HOME/OpenBSD/3.8/i386
```

### Menyiapkan file-file yang dibutuhkan

Metode berikut dapat digunakan untuk membuat file ISO untuk OpenBSD RELEASE atau *snapshot*nya. Anda dapat menggunakan program sejenis ftp(1) untuk mendownload file-file yang dibutuhkan, atau jika Anda melakukan make release, set RELEASEDIR ke path yang dimaksud.

```
$ cd $HOME/OpenBSD/3.8/i386
$ ftp -n ftp://ftp.openbsd.org/pub/OpenBSD/3.8/i386/*
$ ls
CKSUM          base38.tgz      cdboot          etc38.tgz      index.txt
xetc38.tgz
INSTALL.i386   bsd            cdb             floppy38.fs    man38.tgz
xfont38.tgz
INSTALL.linux  bsd.mp         cdemu38.iso    floppyB38.fs   misc38.tgz
xserv38.tgz
INSTALL.dbr    bsd.rd         cdrom38.fs     floppyC38.fs   pxeboot
xshare38.tgz
MD5            cd38.iso       comp38.tgz     game38.tgz     xbase38.tgz
```

### Membuat file ISO

Pada artikel ini, kita akan menggunakan mkhybrid(1), selain itu dapat juga digunakan mkisofs

yang terdapat pada ports (7).

```
$ cd $HOME
$ sudo mkhybrid \
  -r \
  -b 3.8/i386/cdrom38.fs \
  -c boot.catalog \
  -o openbsd-x86-3.8.iso \
  OpenBSD/
```

Oh, hampir saja lupa.. Alangkah baiknya jika Anda membeli CD OpenBSD asli, memasyarakatkan OpenBSD ke rekan-rekan yang lain, atau memberikan donasi ke proyek pengembangan OpenBSD, lihat <http://openbsd.org/goals.html#funding>

## SCP (secure copy)

*author: hky*

Makin banyak daemon yang jalan, makin banyak port yang terbuka, , berarti makin banyak perhatian anda yang akan tersita untuk memastikan semuanya dalam keadaan baik baik saja. Kenapa harus menghidupkan ftpd kalau seandainya ssh dengan scp sudah bisa melakukan fungsi transfer file. Memang dibandingkan ftp, dari segi kecepatan scp sedikit lebih lambat dibandingkan dengan ftp, karena ada nya process enkripsi dalam pengiriman data.

Ok.. sekarang bagaimana menggunakan scp ?

```
$scp "dari" "ke" (prinsip dasar)
$scp /path/to/filename.php ballysta@remote.ho.st:/var/www/ (transfer dari
local ke remote)
$scp ballysta@remote.ho.st:/var/www/filename.php ./ (transfer dari remote ke
local)
$scp -r ballysta@remotel.ho.st:/var/www/*
ballysta@remote2.ho.st:/srv/www/htdocs/
  (transfer seluruh isi direktori /var/www/ dari remote host 1 ke direktori
/srv/www/htdocs/ remote host 2)
```

## Multiple session login dengan background-control dan screen

Ingin lebih cepat dan efektif bekerja? nah, silahkan terapkan beberapa panduan singkat berikut untuk menghemat waktu anda ketika anda sedang *login* ke *remote-terminal*:

- Install dan pelajari program screen (1)
  - *screen* memungkinkan kita untuk menjalankan *multiple login session* didalam sistem
  - fitur *attach* dan *detach* memungkinkan kita untuk meletakkan/menyimpan sesi tersebut dan memanggilnya kembali apabila kita *login* kembali ke sistem (contoh: ketika tiba-tiba saja koneksi ke *remote-login* terputus, apabila kita *login* kembali ke sistem, kita dapat

meminta **screen** untuk meletakkan proses yang terputus tadi dan melanjutkan aktivitas kembali), jangan lupa pasang password agar sesi yang anda jalankan tidak bisa diambil oleh orang lain

- **screen** dapat dikonfigurasi lewat file `.screenrc`, contoh file ini dapat anda lihat di <http://dotfiles.com/>
- *public-key-authentication, ssh-agent* dan *EscapeChar*
  - *public-key-authentication* memudahkan kita dalam melakukan proses autentikasi, meningkatkan keamanan dengan pertukaran kunci dan sentralisasi autentikasi yang lebih baik (karena otak kita terbatas), silahkan baca `ssh-keygen(1)`
  - letakkan proses **ssh** kedalam background dengan menekan `~+CTRL-Z`, panggil kembali apabila anda ingin kembali ke sistem
  - pelajari dan gunakan `ssh-agent(1)`, potongan `.profile` berikut mungkin dapat sedikit membantu,

```
# ~/.profile || ~/.bash_profile
# load ssh key kedalam memori
function ssha {
    eval `/usr/bin/ssh-agent -s`
    /usr/bin/ssh-add
}
# unload ssh-key
function sshk {
    eval `/usr/bin/ssh-agent -k`
}
```

Beberapa artikel menarik yang bisa anda kunjungi: <http://www.hackinglinuxexposed.com/articles>

## Mount File ISO

author: hky

```
FreeBSD
#mdconfig -a -t vnode -f file.iso -u 4;
#mount -t cd9660 /dev/md4 /mountpointnya
```

```
OpenBSD dan NetBSD
#vnconfig svnd0 file.iso
#mount -t cd9660 /dev/svnd0c /mountpointnya
```

## Stripping comments for quick and easy viewing

put this alias on your startup script:

```
alias cstrip='sed -e '\''/^#/d\'\' -e '\''/^$/d\'\''
```

those oneliner will strip leading # and empty character on every line it finds, run it as: `cstrip <text_file_you_want_to_strip>`

From:

<http://wiki.corebsd.or.id/> - **CoreBSD Wiki**

Permanent link:

<http://wiki.corebsd.or.id/doku.php/knowledgebase>

Last update: **2025/10/25 17:09**

