คำสั่งพื้นฐานการใช้งานระบบปฏิบัติการ UNIX (Linux)

คำสั่งเกี่ยวกับ File System

Path

Path คือที่อยู่ของ File หรือ Directory ในระบบ Unix แบ่งเป็น 2 ชนิดคือ

Absolute Path อ้างอิงจาก / (เรียกว่า root directory เป็น Directory เริ่มต้นของระบบ File) ตัวอย่างการ /etc/passwd เป็นที่อยู่ของ File ที่เก็บรายละเอียดของผู้ใช้งานในระบบ

Relative Path อ้างอิงจาก Directory ที่ทำงานอยู่ปัจจุบัน (Working Directory) การอ้างอิงแบบ Relative มีสัญลักษณ์แทน Directory ปัจจุบันเป็น . (จุด)และ Directory ที่อยู่เหนือขึ้นไปหนึ่งระดับ (Parent Directory) แทนด้วย .. (จุดสองจุดติดกัน)

Wildcard เป็นสัญลักษณ์แทนชื่อ File หรือ Directory

* แทนตัวอักษรกี่ตัวก็ได้ (ชื่อ File เป็นอะไรก็ได้)

? แทน 1 ตัวอักษร

Working Directory การอ้างอิงถึง File หรือ Directory ที่ไม่เป็นแบบ Absolute Path จะเป็นการอ้างอิงกับ Working Directory เสมอ

การแสดง Working Directory

หลังจาก login เข้าสู่ระบบแล้ว จะแสดง Command Prompt ของ Shell ถ้าต้องการแสดง Directory ปัจจุบันที่ทำงานอยู่ใช้คำสั่ง pwd (Print Working Directory) ได้ผลลัพธ์ดังนี้

```
#pwd
/root
#
```

แสดงว่าปัจจุบันทำงานอยู่ที่ Directory /root (Working Directory)

การเปลี่ยน Working Directory

ใช้คำสั่ง cd (Change Directory) แล้วตามด้วยที่อยู่ของ Directory เช่นถ้าต้องการเปลี่ยนไปทำงานที่ Directory /etc ใช้คำสั่ง cd /etc แล้วตามด้วยคำสั่ง pwd เพื่อแสดง Directory ปัจจุบันที่เปลี่ยนไป

```
#cd /etc/rc.d
#pwd
/etc/rc.d
```

การแสดงรายชื่อ File และ Directory ย่อย

ใช้คำสั่ง ls (List) แล้วตามด้วยที่ Path ของ Directory ที่ต้องการแสดงรายละเอียด ถ้าไม่ใส่จะหมายถึงแสดงรายละเอียดของ Directory ปัจจุบันที่ทำงานอยู่

```
#ls
init.d/  rc*  rc0.d/  rc1.d/  rc2.d/  rc3.d/  rc4.d/  rc5.d/
rc6.d/  rc.local*  rc.modules*  rc.sysinit*
```

Option ที่ใช้งาน -l ใช้แสดงรายละเอียดทั้งหมด -a ใช้แสดง hidden File หรือ Directory โดย Hidden File และ Hidden Directory ใน Unix คือ File หรือ Directory ที่มีชื่อขึ้นต้นด้วย . (จุด)

การจัดการ File

การสร้าง File สามารถทำได้หลายวิธี ถ้าต้องการ File ขนาด 0 byte สามารถใช้คำสั่ง touch แล้วตามด้วยชื่อ File

```
#touch unix.txt
#ls -l
total 0
-rw-r--r--    1 root     root            0 Jan 15 07:19 unix.txt
```

การ copy File ใช้คำสั่ง cp ชื่อ File ต้นฉบับ ชื่อ File สำเนา

```
#cp unix.txt  data.txt
#ls -l
total 0
-rw-r--r--    1 root     root            0 Jan 15 07:20 data.txt
-rw-r--r--    1 root     root            0 Jan 15 07:19 unix.txt
```

การย้าย File (เปลี่ยนชื่อ File คือการย้าย File ไว้ใน Directory เดิมแต่ใช้ชื่อต่างจากชื่อเดิม) ใช้คำสั่ง mv

```
#mv unix.txt linux.txt
#ls -l
total 0
-rw-r--r--    1 root     root            0 Jan 15 07:20 data.txt
-rw-r--r--    1 root     root            0 Jan 15 07:19 linux.txt
```

การลบ File ใช้คำสั่ง rm

```
#rm linux.txt
rm: remove regular empty file `linux.txt'? y
#ls -l
-rw-r--r--    1 root     root            0 Jan 15 07:20 data.txt
```


การจัดการเกี่ยวกับ Directory

สร้าง Directory mkdir <Directory Name>

```
#mkdir web
#ls -l
total 4
drwxr-xr-x    2 root     root         4096 Jan 15 07:17 web/
```

การสร้าง Directory แบบหลายๆ ชั้น ใส่ Option -p (ตามปกติเราสามารถสร้าง Directory ได้ครั้งละชั้น)

```
#mkdir -p /user/web/data
```

ย้าย Directory (เปลี่ยนชื่อ) mv <Source Path> <Destination Path>

```
#mv web ftp
#ls -l
total 4
drwxr-xr-x    2 root     root         4096 Jan 15 07:17 ftp/
```

ลบ Directory rmdir <Directory Name>

```
#rmdir ftp
#ls -l
total 0
```

คำสั่งแสดงข้อมูลใน File

คำสั่ง cat ตามด้วยชื่อ File

```
#cat /etc/passwd
```

เป็นการแสดงรายระเอียดของผู้ใช้งานในระบบ การใช้คำสั่ง cat จะแสดงข้อมูลทั้งหมดออกที่หน้าจอ ถ้าข้อมูลใน File มีเกินกว่าจำนวนบรรทัดบนหน้าจอเราจะเห็นเฉพาะหน้าจอสุดท้าย

```
#cat /var/log/messages
```

คำสั่ง more ตามด้วยชื่อ File เป็นการแสดงข้อมูลใน File ออกทางหน้าจอครั้งละหน้าจอ กด Spacebar ถ้าต้องการแสดงข้อมูลหน้าจอถัดไป หรือกด Enter เพื่อแสดงข้อมูลบรรทัดต่อไป กด q เพื่อออกจากคำสั่งก่อนการแสดงผลจะจบ ระหว่างการแสดงถ้าต้องการค้นหาข้อความใดสามารถพิมพ์ / แล้วตามด้วยข้อความนั้นแล้วกด Enter

```
#more /var/log/messages
```

(คำสั่ง less ทำงานได้เช่นเดียวกันกับคำสั่ง more แต่มีรายละเอียดมากกว่า)

```
#less /var/log/messages
```


คำสั่งที่ใช้นำจำนวนตัวอักษร จำนวนบรรทัด จำนวนคำใน File

wc ตามด้วยชื่อ File

```
#wc /etc/passwd
     39      91    1886 /etc/passwd
```

จากผลลัพธ์ File /etc/passwd มี 39 บรรทัด 91 คำ และ 1886 ตัวอักษรตามลำดับ

คำสั่งกรองข้อมูล grep

grep <ข้อความที่ต้องการ> <ชื่อ File>

ต้องการหาคำว่า root จาก File /etc/passwd คำสั่งที่เรียกใช้คือ

```
#grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

ผลลัพธ์ที่ได้จะแสดงเฉพาะบรรทัดที่อยู่ใน File /etc/passwd ซึ่งมีคำว่า root

ต้องการหาคำว่า DocumentRoot จาก File /etc/httpd/conf/httpd.conf

```
#grep DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot /var/www/html
```

แสดงผลลัพธ์จากคำสั่ง sysctl เฉพาะบรรทัดที่มีคำว่า forward

```
[root@garfield root]# sysctl -a |grep forward
net.ipv4.conf.eth1.mc_forwarding = 0
net.ipv4.conf.eth1.forwarding = 1
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 1
net.ipv4.ip_forward = 1
```

คำสั่ง sort ใช้สำหรับค้นหา File

sort <filename>

```
#sort /etc/passwd
adm:x:3:4:adm:/var/adm:/bin/sh
alias:x:81:11:alias user:/var/qmail/alias:/bin/true
amavis:x:97:502:Anti Virus Checker:/var/virusmails:/bin/false
apache:x:72:72:system user for apache:/var/www:/bin/sh
bin:x:1:1:bin:/bin:/bin/sh
clamav:x:90:90:system user for clamav:/var/lib/clamav:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
ftp:x:76:76:system user for proftpd:/var/ftp:/bin/false
games:x:12:100:games:/usr/games:/bin/sh
```

คำสั่ง find ใช้สำหรับค้นหา File

find <start Directory> -name <filename>

ค้นหา file ที่ชื่อ wget เริ่มต้นจาก root Directory /

```
#find / -name wget
find: /mnt/floppy: Input/output error
find: /mnt/cdrom: Input/output error
/usr/bin/wget
```

filename สามารถกำหนดโดยใช้ wildcard (* ?) แทนได้แต่ให้ครอบไว้ด้วย ‘ (single quote)

ค้นหา file ที่มีชื่อขึ้นต้นด้วย syslog แล้วตามด้วยอะไรก็ได้ เริ่มต้นจาก root Directory /

```
#find / -name 'syslog*'
/etc/sysconfig/syslog
/etc/rc.d/init.d/syslog
/etc/logrotate.d/syslog
/etc/syslog.conf
/etc/webmin/syslog
```

ดูรายละเอียดเพิ่มเติมจากคำสั่ง man find

คำสั่ง ln สำหรับสร้าง Link

Link แบ่งเป็น 2 ชนิดคือ

hard link ไม่สามารถ link ข้าม file system เป็นการชี้ไปที่ data block ของ file นั้นๆ ถ้าลบ file ต้นฉบับยังสามารถเข้าถึงข้อมูลของ file นั้นผ่าน hard link ได้

symbolic link สามารถ link ข้าม file system ได้ เป็นการชี้ไปยังชื่อ file ต้นฉบับ ถ้าลบ file ต้นฉบับ จะไม่สามารถเข้าถึงข้อมูลใน file นั้นได้

ln <Source> <Link>

```
# cp /etc/hosts ./
# ls -l
total 4
-rw-r--r--    1 root     root           93 Jan 15 09:34 hosts

# ln hosts myhosts
# ls -l
total 8
-rw-r--r--    2 root     root           93 Jan 15 09:34 hosts
-rw-r--r--    2 root     root           93 Jan 15 09:34 myhosts

# cat hosts
195.168.1.254   garfield.info.com
# cat myhosts
195.168.1.254   garfield.info.com
```

```
# rm hosts
rm: remove regular file `hosts'? y

# ls -l
total 4
-rw-r--r--    1 root     root           93 Jan 15 09:34 myhosts
# cat myhosts
195.168.1.254   garfield.info.com
```

ถ้าต้องการสร้าง symbolic link ให้ใช้ option -s

```
# ls -l /var/log/messages
-rw-r-----    1 root   adm  638744 Jan 15 09:23 /var/log/messages

# ln -s /var/log/messages systemlog
# ls -l
total 0
lrwxrwxrwx    1 root   root    17 Jan 15 09:23 systemlog ->
/var/log/messages
# more systemlog
```


Pipe

คำสั่งที่เรียกใช้ผ่าน Shell ของ Unix สามารถส่งผลลัพธ์จากคำสั่งหนึ่งไปเป็น input ของอีกคำสั่งหนึ่งได้ เราเรียกวิธีการส่งนี้ว่า Pine มีสัญลักษณ์เป็น |

คำสั่ง ls ใช้สำหรับแสดงชื่อ File และ Directory และคำสั่ง more ใช้แสดงผลลัพธ์ครั้งละหน้าจอ ถ้าต้องการให้ผลลัพธ์จากคำสั่ง ls ส่งไปเป็น input สำหรับคำสั่ง more เรียกใช้คำสั่งดังนี้

```
#ls -l /etc | more
```

Redirect

Unix ประกอบด้วย Standard Input คือ Keyboard โดย Standard Output และ Standard Error คือ Terminal ถ้าไม่ได้กำหนดเป็นอย่างอื่นคำสั่งที่ใช้จะรับ Input จาก Keyboard และส่ง Output และ Error ออกที่ Terminal แต่ถ้าต้องการเปลี่ยนทิศทาง Input หรือ Output ให้เป็นอย่างอื่น สามารถทำได้โดยการ Redirect ใช้สัญลักษณ์เป็น > หรือ <

Standard Output ใช้เครื่องหมาย > เป็นการส่ง Output ออกไปเป็น File ใหม่ ถ้ามีอยู่แล้ว File นั้นจะถูกเขียนทับไป ถ้าต้องการให้ Output ถูกบันทึกต่อท้าย File ที่มีอยู่ให้ใช้เป็นเครื่องหมาย >>

ตัวอย่างเช่นถ้าต้องการเก็บ Output จากคำสั่ง ls -l /etc ไว้เป็น File

```
#ls -l /etc > /tmp/output
```

ที่ Terminal จะไม่แสดงผลลัพธ์ใดๆ ออกมาแต่จะมีการสร้าง File ขึ้นใหม่มีชื่อว่า output เก็บไว้ใน Directory ปัจจุบัน (สามารถใช้คำสั่ง cat หรือ more เพื่อแสดงข้อมูลใน file)

Standard Input ใช้เครื่องหมาย < เป็นการรับค่า Input จาก File

Standard Error แทนด้วยหมายเลข 2 ถ้าต้องการส่งข้อมูลจาก Standard Error เก็บลง File (ตามปกติ ข้อมูล Standard Error จะแสดงออกทาง Terminal) ใช้เป็น command 2>filename

```
# ls /nofile
ls: /nofile: No such file or directory
```

การ Redirect Standard Error ลง File /tmp/error ทำได้ดังนี้

```
# ls /nofile 2>/tmp/error
```

สังเกตที่ Terminal จะไม่แสดง error แต่จะเก็บลงใน File /tmp/error

คำสั่ง fdisk ใช้สำหรับแบ่ง Partition

fdisk <device>

```
[root@tom root]# fdisk /dev/hda

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```



```
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 553 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1       261   2096451    7  HPFS/NTFS
/dev/sda2           262       553   2345490    5  Extended
/dev/sda5           459       553    763056   83  Linux
/dev/sda6           262       427   1333332   83  Linux
/dev/sda7           428       444    136521   82  Linux swap

Partition table entries are not in disk order

Command (m for help):
```

```
Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (445-553, default 445): 445
Last cylinder or +size or +sizeM or +sizeK (445-458, default 458):
458

Command (m for help):
```

```
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 553 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1       261   2096451    7  HPFS/NTFS
/dev/sda2           262       553   2345490    5  Extended
/dev/sda5           459       553    763056   83  Linux
/dev/sda6           262       427   1333332   83  Linux
/dev/sda7           428       444    136521   82  Linux swap
/dev/sda8           445       458    112423+  83  Linux

Partition table entries are not in disk order

Command (m for help):
```

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

Calling ioctl() to re-read partition table.
Re-read table failed with error 16: Device or resource busy.
Reboot your system to ensure the partition table is updated.

WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
#reboot
```

คำสั่ง mkfs ใช้สำหรับ Format File System

```
# mkfs /dev/sda8
mke2fs 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
28112 inodes, 112423 blocks
5621 blocks (5.00%) reserved for the super user
First data block=1
14 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

คำสั่ง fsck ใช้ตรวจสอบ File System ควรตรวจสอบ File System ที่ไม่ได้ใช้งานอยู่ (ไม่ได้ mount)

```
# fsck /dev/sda8
Parallelizing fsck version 1.18 (11-Nov-1999)
e2fsck 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
/dev/sda8: clean, 11/28112 files, 3568/112423 blocks
```



```
#fsck  /dev/sda5
Parallelizing fsck version 1.18 (11-Nov-1999)
e2fsck 1.18, 11-Nov-1999 for EXT2 FS 0.5b, 95/08/09
/dev/sda5 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/dev/sda5 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda5: 15599/95424 files (9.3% non-contiguous), 174513/190764
blocks
```

คำสั่ง mount เพื่อใช้งาน File System

mount <filesystem> <mount point>

```
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda6              1312340   1229048     16628  99% /
/dev/sda5               751048    686044     26852  97% /user
# mkdir /website
# mount /dev/sda8 /website
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda6              1312340   1229084     16592  99% /
/dev/sda5               751048    686044     26852  97% /user
/dev/sda8               108868        13    103234   1% /website
```

เพิ่มลงท้าย file /etc/fstab เพื่อให้ file system /dev/sda8 ถูก mount ทุกครั้งที่เปิดเครื่อง Server

```
/dev/sda8       /website         ext2    defaults        0 0
```

File Compression

คำสั่ง tar

tar cvf <file.tar> <source path>

```
# mkdir backup
# cd backup
# tar cvf etc.tar /etc/
# ls -l
total 9048
-rw-r--r--    1 root     root      9246720 Jan 14 12:57 etc.tar
```

คำสั่ง gzip/gunzip

gzip <filename>

gunzip <filename.gz>

```
# gzip etc.tar
# ls -l
total 1636
-rw-r--r--    1 root     root     1668422 Jan 14 12:57 etc.tar.gz
```

```
# ls -l
total 1636
-rw-r--r--    1 root     root     1668422 Jan 14 12:57 etc.tar.gz
# gunzip etc.tar.gz
# ls -l
total 9048
-rw-r--r--    1 root     root      9246720 Jan 14 12:57 etc.tar
```

```
# tar xvf etc.tar
# ls -l
total 9052
drwxr-xr-x   70 root     root         4096 Jan 14 12:40 etc/
-rw-r--r--    1 root     root      9246720 Jan 14 12:57 etc.tar
```


คำสั่ง zip/unzip

zip <filename>

unzip <filename.zip>

```
$ zip etc.zip /etc/*
$ ls -l
total 460
-rw-r--r--    1 mong     users      465312 Jan 14 13:51 etc.zip
```

```
$ ls -l
total 460
-rw-r--r--    1 mong     users      465312 Jan 14 13:51 etc.zip
$ unzip etc.zip
$ ls -l
total 464
drwxr-xr-x   48 mong     users        4096 Dec 28 17:31 etc
-rw-r--r--    1 mong     users      465312 Jan 14 13:51 etc.zip
```

คำสั่ง ssh (Secure SHell) ใช้สำหรับการ Remote Login เข้าใช้งานเครื่องผ่านเครือข่าย TCP/IP แบบ Secure Communication ข้อมูลระหว่าง SSH Client กับ SSH Server จะถูกเข้ารหัส (Encryption) ไว้

ssh -l <loginname> <hostname>

```
# ssh -l user 192.168.3.1
user@192.168.2.1's password:
[user@kitty user]$
```

คำสั่ง telnet ใช้สำหรับการ Remote Login เข้าใช้งานเครื่อง Server ผ่านทางเครือข่าย TCP/IP

telnet <remote hostname>

```
[mong@sim mong]$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.


Digital UNIX (gw-server) (ttyp0)

login: garfield
Password:

$
```

คำสั่ง ftp ใช้สำหรับการ Download/Upload File

ftp <hostname>

ตัวอย่างการ Upload File webmin-1.030-1.noarch.rpm จากเครื่อง hplinux ไปยังเครื่อง FTP Server tom.info.com (195.168.3.254)

```
# ls -la
total 6368
drwxr-xr-x    2 root     root         4096 Jan 15 06:42 ./
drwx------   25 root     root         4096 Jan 14 15:33 ../
-rw-r--r--    1 root     root      6500238 Jan 15 06:42 webmin-1.030-
1.noarch.rpm
```


การใช้งานคำสั่ง ftp

```
[root@hplinux src]# ftp 195.168.3.254
Connected to 195.168.3.254.
220 ProFTPD 1.2.5 Server (ProFTPD Default Installation) [tom.info.com]
KERBEROS_V4 rejected as an authentication type
Name (195.168.3.254:root): user1
331 Password required for user1.
Password:
230 User user1 logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (195,168,3,254,4,7).
150 Opening ASCII mode data connection for file list
drwx------   2 user1    user1        4096 Jan  2 02:04 tmp
226-Transfer complete.
226 Quotas off
ftp> put  webmin-1.030-1.noarch.rpm
local: webmin-1.030-1.noarch.rpm remote: webmin-1.030-1.noarch.rpm
227 Entering Passive Mode (195,168,3,254,4,13).
150 Opening BINARY mode data connection for webmin-1.030-1.noarch.rpm
226 Transfer complete.
6500238 bytes sent in 20 seconds (3.2e+02 Kbytes/s)
ftp> ls
227 Entering Passive Mode (195,168,3,254,4,15).
150 Opening ASCII mode data connection for file list
drwx------   2 user1    user1        4096 Jan  2 02:04 tmp
-rw-r--r--   1 user1    user1     6500238 Jan 15 00:55 webmin-1.030-
1.noarch.rpm
226-Transfer complete.
226 Quotas off
ftp> quit
221 Goodbye.
#
```

Anonymous FTP

การเข้า Download File โดยไม่ต้องมี User name และ Password บนเครื่อง FTP Server โดยใช้ User Name เป็น ftp หรือ anonymous และ Password เป็น Email Address

ตัวอย่างการเข้า Download โปรแกรม winzip จาก FTP Server ftp.chula.ac.th

```
[mong@sim mong]$ mkdir download
[mong@sim mong]$ cd download
[mong@sim download]$ ftp ftp.chula.ac.th
Connected to ftp1.it.chula.ac.th.
220 ftp1.it.chula.ac.th NcFTPd Server (free educational license)
ready.
Name (ftp.chula.ac.th:mong): ftp
331 Guest login ok, send your complete e-mail address as password.
230-You are user #10 of 50 simultaneous users allowed.
230-
230 Logged in anonymously.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (161,200,192,3,205,191)
150 Data connection accepted from 161.200.80.73:1051; transfer
starting.
dr-xr-xr-x   2 ftpuser  ftpusers       512 Jun 15  2002 bin
dr-xr-xr-x   2 ftpuser  ftpusers       512 Jun 15  2002 etc
drwxrwxr-x  11 ftpuser  ftpusers       512 Dec 11 11:58 pub
226 Listing completed.
ftp> cd pub/windows/compression
250 "/pub/windows/compression" is new cwd.
ftp> get winzip80.exe
local: winzip80.exe remote: winzip80.exe
227 Entering Passive Mode (161,200,192,3,205,192)
150 Data connection accepted from 161.200.80.73:1052; transfer
starting for winzip80.exe (1259448 bytes).
226 Transfer completed.
1259448 bytes received in 0.52 secs (2.4e+03 Kbytes/sec)
ftp> quit
221 Goodbye.
[mong@sim download]$ ls -l
total 1236
-rw-r--r--    1 mong     users     1259448 Jan 14 16:34 winzip80.exe
```


Process

Process คือโปรแกรมที่กำลังทำงานอยู่ (มีการใช้งาน Memory และ CPU) โปรแกรมที่เก็บไว้ใน Hard Disk เป็นเพียงชุดคำสั่งที่ยังไม่ได้ทำงาน แต่ถ้าถูกเรียกใช้งานจะมีการจอง Resource (Memory CPU ...) แล้วนำโปรแกรมไปทำงาน ดังนั้นโปรแกรมเดียวกันสามารถถูกเรียกใช้งานเป็น Process ได้หลายๆ Process

Process แต่ละ Process มีหมายเลข Process เรียกว่า Process ID (PID) ซึ่งเป็นตัวเลขที่ไม่ซ้ำกัน เพื่อใช้อ้างอิงและควบคุมการทำงานของ Process ต่างๆ โดยผู้ที่สามารถควบคุม Process ได้ต้องเป็นผู้ที่สร้าง Process นั้นขึ้นมา (เป็นเจ้าของ Process) โดยมีข้อยกเว้นว่า root สามารถควบคุม Process ทุกๆ Process ในระบบ

คำสั่งที่ใช้แสดงรายละเอียดเกี่ยวกับ Process ใช้คำสั่ง ps

```
# ps
  PID TTY          TIME CMD
23633 pts/1    00:00:00 bash
27171 pts/1    00:00:00 ps
```

```
# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Dec17 ?        00:00:04 init [3]
root         2     1  0 Dec17 ?        00:00:00 [keventd]
root         3     1  0 Dec17 ?        00:00:00 [kapmd]
root         4     1  0 Dec17 ?        00:00:00 [ksoftirqd_CPU0]
root         5     1  0 Dec17 ?        00:00:06 [kswapd]
```

.... ผลลัพธ์ที่เหลือถูกละไว้ ....

(ถ้าต้องการแสดงผลครั้งละหน้าจอคำสั่งที่เรียกใช้เป็น ps -ef | more)

Option e กำหนดให้แสดง Process ทั้งหมดในระบบ Option f แสดงรายละเอียดทั้งหมด จากผลลัพธ์แต่ละ Column มีรายละเอียดดังนี้

UID แสดงเจ้าของ Process

PID แสดง Process ID

PPID แสดง Parent Process ID (Process ที่เป็นคนสร้าง Process นี้ขึ้นมา)

CMD คำสั่งที่ถูกเรียกให้ทำงานเป็น Process

Job Control (การควบคุม Process)

การทำงานของ Process แบ่งเป็น 2 ชนิด

การทำงานแบบ Foreground

การทำงานจะติดต่อกับผู้ใช้งาน เมื่อทำงานแบบ Foreground จะต้องทำงานให้จบก่อนถึงสามารถเรียกใช้คำสั่งอื่นต่อไปได้ สังเกตได้ระหว่างที่ทำงานจะไม่เห็น Command Prompt แต่เมื่อ Process ทำงานจบแล้วจะกลับมาที่ Command Prompt

```
#find / -name passwd
```

เป็นคำสั่งที่ใช้ค้นหาชื่อ File ที่ตรงกับคำว่า passwd โดยเริ่มค้นหาจาก / (root Directory) ผลลัพธ์ที่ได้แสดงออกที่หน้าจอ Terminal (สังเกตว่าระหว่างทำงานจะไม่สามารถทำงานอื่นได้ ต้องรอให้คำสั่งนี้ทำงานเสร็จก่อน)

การทำงานแบบ Background

การทำงานจะไม่ติดต่อกับผู้ใช้งาน เมื่อเรียกใช้คำสั่งแล้วจะกลับสู่ Command Prompt ถ้าต้องการให้คำสั่งที่เรียกใช้ทำงานแบบ Background ให้เพิ่มเครื่องหมาย & ต่อท้ายคำสั่งนั้นๆ (การเรียกใช้งานแบบ Background ควรจัดการเรื่องของ Input และ Output ให้ Input รับจาก File หรือกำหนด Option ของคำสั่งให้เรียบร้อย และ Output ส่งออกไปที่ File เพราะระหว่างการทำงานจะไม่สามารถรับข้อมูลจากผู้ใช้งานได้)

```
#find / -name passwd >/tmp/output &
```

ผลลัพธ์ที่ได้เก็บไว้ใน File /tmp/output และทำงานแบบ Background ดังนั้นเมื่อพิมพ์คำสั่งแล้วจะกลับสู่ Command Prompt

การควบคุม Process ที่ทำงานแบบ Background

ขั้นตอนแรกต้องหา Process ID ของ Process ที่ต้องการควบคุมก่อน เช่นหา Process ID ของ named (Service ของ DNS Server)

```
#ps -ef|grep named
named     1236     1  0 Dec17 ?        00:00:00 named -u named
```

Process ID อยู่ใน Column ที่ 2 ในที่นี้เป็นหมายเลข 1236
จากนั้นถ้าต้องการจบการทำงานของ Process ใช้คำสั่ง kill -<Signal Number> <Process ID> เพื่อส่งสัญญาณไปควบคุม Process

```
#kill -1 1236
```

-1 คือสัญญาณ HUP (Hang Up) บอกให้ Process ของ named เริ่มอ่านค่าเริ่มต้นการทำงาน (Configuration) ขึ้นมาใหม่
-9 คือสัญญาณ Terminate บอกให้ Process จบการทำงานแบบไม่มีเงื่อนไข
ถ้าไม่กำหนด Signal Number จะเป็นการสั่งให้ Process จบการทำงานแบบปกติ

คำสั่ง top

ใช้แสดงรายการของ Process ทั้งหมดในระบบ

```
# top

  1:15pm  up 7 days, 15:08,  3 users,  load average: 0.17, 0.12, 0.08
136 processes: 134 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  0.5% user,  0.3% system,  0.1% nice, 98.8% idle
Mem:   256956K av,  240868K used,   16088K free,       0K shrd,   13540K buff
Swap:  522072K av,   25320K used,  496752K free                  106956K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
 5865 root      11  -1 62504  12M  2344 S <   0.3  5.0 213:03 X
29908 root      12   0  1088 1088   816 R     0.3  0.4   0:00 top
29495 root       9   0  2040 2000  1688 S     0.1  0.7   0:00 sshd
29737 root      19  19 10316  10M  8208 R N   0.1  4.0   0:08 krozat.kss
    1 root       8   0   172  140   108 S     0.0  0.0   0:04 init
    2 root       9   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root       9   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      19  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root       9   0     0    0     0 SW    0.0  0.0   0:06 kswapd
    6 root       9   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    7 root       9   0     0    0     0 SW    0.0  0.0   0:00 kupdated
    8 root      -1 -20     0    0     0 SW<   0.0  0.0   0:00 mdrecoveryd
   12 root       9   0     0    0     0 SW    0.0  0.0   0:00 scsi_eh_0
   88 root       9   0   316  200   160 S     0.0  0.0   0:00 devfsd
  184 root       9   0     0    0     0 SW    0.0  0.0   0:00 khubd
```


Editor

การใช้เครื่องมือเฉพาะของ Linux เป็นเครื่องมือที่ช่วยอำนวยความสะดวกในการกำหนดค่าเริ่มต้นการทำงานของ Linux Distribution ต่างๆ มีข้อดีคือช่วยให้ทำงานได้รวดเร็ว แต่ข้อเสียคือใน Linux Distribution ต่างๆ เครื่องมือที่ใช้อาจไม่เหมือนกัน และเวลาเกิดปัญหาบางอย่างแล้วจะยากต่อการหาสาเหตุและแก้ไข

ค่าเริ่มต้นต่างๆ ของ Linux หรือ Unix เก็บไว้เป็น Text File เครื่องมือต่างๆ เป็นเพียงวิธีการเข้าไปแก้ไขข้อมูลใน Text File เพื่อให้ผู้ใช้งานสามารถเรียกใช้ได้อย่างสะดวก ดังนั้นเราควรเรียนรู้ทั้งสองวิธี คือการเข้าใช้งานผ่านเครื่องมือในสภาวะปกติ และสามารถเข้าใช้งานผ่าน Text File ได้เมื่อจำเป็น เครื่องมือที่สำคัญสำหรับการเข้าแก้ไข Text File คือ Editor ซึ่งมีให้ใช้งานอยู่ 2 ชนิดคือ pico และ vi

pico เป็น Editor ที่ใช้งานง่ายคล้ายกับ Editor ที่ใช้งานบน DOS ทั่วๆ ไป ถ้าต้องการพิมพ์หรือแก้ไขหรือลบข้อความสามารถทำได้ทันที

vi แบ่งโหมดการทำงานออกเป็น 2 โหมดหลักๆ คือ โหมดคำสั่ง (Command Mode)และ โหมดแก้ไข (Edit Mode)
โดยความสามารถและความแพร่หลายแล้ว vi มีสูงกว่า pico แต่ pico ใช้งานง่ายกว่า

การใช้งาน pico เบื้องต้น

```
#pico file2.txt
```

ภายในโปรแกรม pico สามารถพิมพ์ข้อความและแก้ไขได้เหมือนกับ Editor ทั่วไป เมื่อต้องการบันทึกใช้คำสั่ง Ctrl+w แล้ว Enter เพื่อบันทึกเป็นชื่อเดิม หรือ เปลี่ยนชื่อ File ก่อนแล้วจึง Enter เป็นการบันทึกเป็นอีกชื่อหนึ่ง ถ้าต้องการออกจาก Editor ใช้คำสั่ง Ctrl+x แล้ว Enter ถ้ายังไม่มีการบันทึกการเปลี่ยนแปลงใน File จะมีการเตือนให้บันทึก ถ้าต้องการบันทึกให้ตอบ y แล้ว Enter ถ้าไม่บันทึกการเปลี่ยนแปลงให้ตอบ n แล้ว Enter

การค้นหาคำใน Editor pico

```
#pico /etc/httpd/conf/httpd.conf
```

ใช้คำสั่ง Ctrl+w แล้วใส่คำที่ต้องการค้นหา จากนั้นกด Enter Cursor จะไปยังตำแหน่งที่พบคำนั้นเป็นรายการแรก ถ้าต้องการแสดงผลลัพธ์ต่อไปให้กด Ctrl+w แล้ว Enter

การใช้งาน vi เบื้องต้น

สร้าง File ใหม่โดยใช้คำสั่ง vi <File Name> ถ้ามี File อยู่แล้วจะเป็นการเปิด File เพื่อแก้ไข

```
#vi file1.txt
```

จะเข้าสู่ Command Mode ถ้าต้องการเพิ่มข้อมูลให้พิมพ์คำสั่ง i (insert) หรือ คำสั่ง a (append) หลังจากพิมพ์คำสั่ง i หรือ a จะไม่แสดงตัว i หรือ a ขึ้นมาแต่จะเข้าสู่ Edit Mode ที่สามารถเพิ่มข้อมูลลงไปใน File ได้
ถ้าต้องการบันทึกข้อมูลให้กลับสู่ Command Mode โดยการกดปุ่ม Esc แล้วทำการบันทึกด้วยการพิมพ์คำสั่ง :w แล้ว Enter และออกจาก vi ด้วยคำสั่ง :q

การค้นหาข้อความใน Editor vi ใน Command Mode ใช้คำสั่ง / แล้วตามด้วยข้อความที่ต้องการค้นหา

```
#vi /etc/httpd/conf/httpd.conf
```

ถ้าต้องการหาคำว่า DocumentRoot ใช้คำสั่ง /DocumentRoot แล้ว Enter จากนั้น Cursor จะไปอยู่ที่ตำแหน่งของคำแรกที่ค้นพบ ถ้าต้องการแสดงผลลัพธ์ต่อไปให้พิมพ์คำสั่ง / แล้ว Enter การค้นหาด้วยคำสั่ง / เป็นการค้นหาจากด้านบนของ File ไปด้านล่าง ถ้าต้องการค้นหาย้อนกลับให้เปลี่ยนจาก / เป็น ?

การไปยังบรรทัดที่ต้องการให้พิมพ์ตัวเลขของบรรทัดแล้วตามด้วย G (จีตัวใหญ่) จากนั้น Cursor จะไปยังบรรทัดที่ต้องการ ถ้าต้องการไปที่บรรทัดสุดท้ายของ File ให้พิมพ์ G (ไม่ต้องใส่หมายเลขบรรทัด)

File System

การนำ File System มาใช้งานต้องทำการ mount คือการกำหนดรายละเอียดของ File System และจุดต่อเชื่อม (Mount Point) Mount Point คือ Directory ที่ใช้บนจุดต่อเชื่อมซึ่งอยู่บน File System หลัก ภายใน Directory นี้ไม่ควรข้อมูลเก็บอยู่ เพราะขณะที่ทำการ mount แล้วจะไม่สามารถเข้าใช้งานข้อมูลที่อยู่ใน Directory ได้ เนื่องจากภายใต้ Directory เป็นข้อมูลของ File System ที่ทำการ mount อยู่

รายละเอียดการ mount File System เก็บอยู่ใน File /etc/fstab

คำสั่งที่ใช้แสดงสถานะการ mount File System คือคำสั่ง mount

```
# mount
/dev/sda1 on / type ext2 (rw)
none on /proc type proc (rw)
none on /proc/bus/usb type usbdevfs (rw)
```

แสดงเนื้อที่ใน File System ต่างๆ ที่ใช้งานอยู่ด้วยคำสั่ง df

```
# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             3.9G  1.7G  2.1G  45% /
```

คำสั่ง mount ใช้ต่อเชื่อม File System เข้ามาใช้งาน รูปแบบของคำสั่ง

mount <Option> <File System> <Mount Point>

mount cdrom มีชื่อ device เป็น /dev/cdrom ไปไว้ที่ mount point /mnt

ตรวจสอบการ mount ด้วยคำสั่ง df

```
# mount /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting
read-only
# ls -l /mnt

# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             3.9G  1.8G  1.9G  49% /
/dev/cdrom            695M  695M     0 100% /mnt
```


ยกเลิกการ mount ด้วยคำสั่ง umount โดย ภายใต้ mount point ต้องไม่มี user ใดใช้งานอยู่

```
# umount /mnt
# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             3.9G  1.8G  1.9G  49% /
```

mount floppy disk (Drive A) มีชื่อ device เป็น /dev/fd0

```
# mount /dev/fd0 /mnt
mount: block device /dev/fd0 is write-protected, mounting read-
only
# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             3.9G  1.8G  1.9G  49% /
/dev/fd0              1.5M  1.2M  222K  85% /mnt
# ls -l /mnt
```

ยกเลิกการ mount ด้วยคำสั่ง umount

```
# umount /mnt
# df
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             3.9G  1.8G  1.9G  49% /
```

ถ้า file system ใดมีรายละเอียดการ mount อยู่ใน file /etc/fstab การ mount ใช้เพียงคำสั่ง mount แล้วตามด้วย mount point เช่น

mount /mnt/cdrom

mount /mnt/floppy

File ที่เก็บค่าเริ่มต้นการทำงานของ Linux ที่สำคัญ

file /etc/inittab

ใช้กำหนดสถานะของระบบเมื่อระบบ Boot ขึ้นมา แบ่งออกเป็นระดับตั้งแต่ 0-6

0	สถานะการปิดเครื่อง
1	Single User Mode เป็นสถานะที่การเข้าใช้งานเครื่องผ่าน Console ไม่มี Password สำหรับการ Login
2	Multi User (ไม่มีการใช้งาน Network File System)
3	Multi User มี Network File System (NFS)
4	สถานะนี้ไม่มีการใช้งาน
5	XWindows ทำงานแบบ Graphic Mode
6	สถานะการ Reboot เครื่อง

ข้อมูลสำคัญใน file

```
id:3:initdefault:
```

เป็นการบอกว่าสถานะการทำงานหลังจาก Boot แล้วให้ทำงานที่ runlevel 3 Full Multi User Mode Runlevel 0 และ 6 ห้ามกำหนดเป็น initdefault เพราะเครื่องจะถูก Shutdown หรือ Reboot ตลอดเวลา ตามปกติแล้ว initdefault จะอยู่ที่ level 3 หรือ level 5

การใช้คำสั่ง linux single ขณะที่เครื่องกำลัง boot (ที่ LILO Prompt) เป็นการกำหนดให้ระบบ Boot เข้าสู่ Single User Mode (level 1) เพื่อข้ามขั้นตอนการ Login ใช้ในกรณีที่ลืม Password สำหรับ root

file /etc/fstab

เป็น file ที่เก็บรายละเอียดการใช้งาน File System ในระบบ มีโครงสร้างดังนี้

```
<device>    <mount point>    <type>    <mount option>    <dump>  <fsck>
```

```
/dev/sda1       /       ext2     exec,dev,suid,rw,usrquota 1 1
none /dev/pts devpts mode=0620 0 0
none /mnt/cdrom supermount dev=/dev/hdc,fs=auto,ro,--
,iocharset=iso8859-1,codepage=850,umask=0 0 0
none /mnt/floppy supermount dev=/dev/fd0,fs=auto,--
,iocharset=iso8859-1,sync,codepage=850,umask=0 0 0
none /proc proc defaults 0 0
/dev/sda5 swap swap defaults 0 0
```

file /etc/hosts

เก็บชื่อเครื่องและ IP Address ของเครื่อง รูปแบบของข้อมูลใน File

<IP Address> <hostname.domainname> <hostname>

ตัวอย่าง

```
127.0.0.1       localhost.localdomain      localhost
172.16.80.2     hplinux.info.com           hplinux
```

ตัวอย่าง

```
search info.com
nameserver 172.16.80.2
```

file /etc/resolv.conf

กำหนดรายละเอียดการติดต่อกับ DNS Server รูปแบบของข้อมูลใน File

search <Domain Name>

nameserver <IP Address DNS Server>

file /var/log/messages

เก็บ System Logging (เหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบ) เวลาเกิดการทำงานผิดพลาดของ Service สามารถเปิดดูรายละเอียดของการทำงานได้จาก file นี้ ตัวอย่างของ System Logging

```
Dec  8 04:12:40 sim named[372]: Lame server on '85.237.144.207.in-
addr.arpa' (in '144.207.IN-ADDR.ARPA'?): [206.74.254.10].53
'DNS2.INFOAVE.NET'
Dec  8 04:12:44 sim named[372]: Lame server on '85.237.144.207.in-
addr.arpa' (in '144.207.IN-ADDR.ARPA'?): [206.74.254.2].53
'DNS4.INFOAVE.NET'
```

คำสั่ง dmesg บอกรายละเอียดของ Hardware ของระบบ

```
# dmesg
Linux version 2.2.16-22 (root@porky.devel.redhat.com) (gcc version
egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)) #1 Tue Aug 22
16:49:06 EDT 2000
Detected 267277 kHz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 532.48 BogoMIPS
Memory: 192232k/196608k available (1048k kernel code, 412k reserved,
```


คำสั่งเกี่ยวกับ Network

Startup และ Shutdown Interface

ifconfig <interfacename> up

ifconfig <interfacename> down

กำหนด IP Address ให้กับ Interface

ifconfig <interface name> inet <ip address> netmask <subnetmask> up

```
# ifconfig eth0 inet 192.168.1.1 netmask 255.255.255.0 up
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe0c:c11b/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3238 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5500 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:295281 (288.3 Kb)  TX bytes:3401104 (3.2 Mb)
          Interrupt:10 Base address:0xe000
```

รายละเอียดของ IP Address ของแต่ละ Interface เก็บอยู่ใน File ifcfg-<Interfacename> เช่น ifcfg-eth0 สำหรับ Interface eth0 ซึ่งอยู่ใน Directory /etc/sysconfig/network-scripts/

คำสั่งสำหรับแสดงสถานะการทำงานของ Network Interface Card และรายละเอียดของ TCP/IP

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:30:6E:0A:06:1B
          inet addr:172.16.80.2  Bcast:172.16.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7286620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51041 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1072907612 (1023.2 Mb)  TX bytes:8360895 (7.9 Mb)
          Interrupt:26 Base address:0x9000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91385 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6477070 (6.1 Mb)  TX bytes:6477070 (6.1 Mb)
```

คำสั่งตรวจสอบสถานะการติดต่อบน TCP/IP

```
# ping 172.16.1.254
PING 172.16.1.254 (172.16.1.254) from 172.16.80.2 : 56(84) bytes of
data.
64 bytes from 172.16.1.254: icmp_seq=1 ttl=128 time=0.603 ms
64 bytes from 172.16.1.254: icmp_seq=2 ttl=128 time=8.17 ms
64 bytes from 172.16.1.254: icmp_seq=3 ttl=128 time=0.544 ms
64 bytes from 172.16.1.254: icmp_seq=4 ttl=128 time=0.560 ms

(กด Ctrl+c เพื่อจบการทำงาน)

--- 172.16.1.254 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3011ms
rtt min/avg/max/mdev = 0.544/2.470/8.176/3.294 ms
```


คำสั่งตรวจสอบการส่งข้อมูลผ่าน Router

```
# traceroute www.chula.ac.th
traceroute to www1.netserv.chula.ac.th (161.200.192.1), 30 hops max, 38 byte packets
 1  161.200.80.227 (161.200.80.227)  0.883 ms  0.736 ms  0.720 ms
 2  f1-0-2-8510-cen32.it.chula.ac.th (161.200.255.154)  2.374 ms  1.519 ms  1.807 ms
 3  g1-0-0-8540-cen59.it.chula.ac.th (161.200.255.230)  0.996 ms  0.975 ms  0.900 ms
 4  f4-0-7513-cen59.it.chula.ac.th (161.200.255.173)  1.912 ms  1.607 ms  1.865 ms
 5  www1.netserv.chula.ac.th (161.200.192.1)  1.497 ms  1.407 ms  1.370 ms
```

คำสั่งแสดง Routing Table ของ Server ใช้คำสั่ง route หรือ netstat -rn

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.0.0      *               255.255.0.0     U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
```

คำสั่งเพิ่ม Default Route

route add default gw <gateway ip address>

```
# route add default gw 192.168.1.254
# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.0.0      *               255.255.0.0     U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default     172.16.1.254        0.0.0.0       UG     0      0        0 eth0
```

File ที่กำหนดรายละเอียดของ Network ของเครื่อง Server เก็บไว้ใน File /etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME="garfield"
DOMAINNAME="info.com"
GATEWAY="172.16.1.254"
GATEWAYDEV="eth0"
FORWARD_IPV4="no"
```

คำสั่งแสดงการติดต่อบน TCP/IP

netstat แสดงเฉพาะการติดต่อที่กำลังใช้งานอยู่

```
# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address          State
tcp        0     52 garfield.info.com:ssh 170.16.1.1:1834     ESTABLISHED
```

ถ้าต้องการแสดงการติดต่อทั้งหมดให้ใส่ Option -a

```
# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address  State
tcp        0      0 localhost.localdom:1024 *:*              LISTEN
tcp        0      0 *:sunrpc                *:*              LISTEN
tcp        0      0 *:10000                 *:*              LISTEN
tcp        0      0 *:ssh                   *:*              LISTEN
tcp        0      0 localhost.localdom:rndc *:*              LISTEN
tcp        0     20 garfield.info.com:ssh   170.16.1.1:1834  ESTABLISHED
udp        0      0 *:1025                  *:*
udp        0      0 *:10000                 *:*
```


การจัดการ Kernel Module

Kernel Module เป็นส่วนประกอบของ Kernel ที่สามารถ Load เพื่อใช้งานได้ตามต้องการ

ตำแหน่งที่เก็บ Kernel Module อยู่ใน Directory /lib/modules/***Kernel Version***/kernel/

```
# ls -l  /lib/modules/2.4.19-16mdk/kernel/
total 24
drwxr-xr-x   26 root     root         4096 Dec 18 22:59 3rdparty/
drwxr-xr-x    3 root     root         4096 Dec 18 22:59 arch/
drwxr-xr-x   26 root     root         4096 Dec 18 23:00 drivers/
drwxr-xr-x   42 root     root         4096 Dec 18 23:00 fs/
drwxr-xr-x   26 root     root         4096 Dec 18 23:00 net/
drwxr-xr-x    9 root     root         4096 Dec 18 23:00 sound/
```

lsmod คำสั่งที่ใช้แสดง Kernel Module ที่กำลังใช้งานอยู่

```
# lsmod
Module                  Size  Used by    Not tainted
ip_vs                  74328   0  (autoclean)
af_packet              13000   0  (autoclean)
ne                      6544   1  (autoclean)
8390                    6192   0  (autoclean) [ne]
8139too                14472   1  (autoclean)
mii                     1152   0  (autoclean) [8139too]
rtc                     6560   0  (autoclean)
```

modprobe เพิ่ม Module

```
# modprobe 8021q
# lsmod
Module                  Size  Used by    Not tainted
8021q                  13832   0  (unused)
ip_vs                  74328   0  (autoclean)
af_packet              13000   0  (autoclean)
ne                      6544   1  (autoclean)
8390                    6192   0  (autoclean) [ne]
8139too                14472   1  (autoclean)
mii                     1152   0  (autoclean) [8139too]
rtc                     6560   0  (autoclean)
```

File ที่เก็บรายการของ Module ที่จะถูกเรียกใช้ /etc/modules.conf

ตัวอย่างการกำหนด Module สำหรับ Network Card

```
alias eth0 8139too
alias eth1 ne
options ne io=0x320 irq=0x5
```

การเปิดปิดเครื่อง Server

คำสั่ง reboot ใช้สำหรับ Restart เครื่อง Server

```
# reboot
```

คำสั่ง halt ใช้สำหรับปิดเครื่อง Server

```
# halt
```

shutdown ใช้สำหรับ shutdown หรือ restart เครื่อง Server (แล้วแต่ Option ที่ใช้)

shutdonw -h now เหมือนกับ halt ใช้สำหรับสั่งปิดเครื่อง

```
# shutdown -h now
```

shutdown -r now เหมือนกับ reboot สั่ง restart เครื่อง

```
# shutdown -r now
```

การ reboot เครื่องสามารถทำได้โดยไม่ต้อง login เป็น root ที่ Console กดปุ่ม Ctrl+Alt+Del

การเพิ่ม Group

ใช้คำสั่ง groupadd

groupadd -g <groupid> <groupname>

```
#groupadd -g 2000 admin
```

ทำได้โดยเข้าไปเพิ่มข้อมูลในต่อท้าย file /etc/group

โดยรูปแบบของ group ที่เพิ่มมีรูปแบบดังนี้

<Groupname>::<Groupid>:

ตัวอย่างเช่นเพิ่ม group ชื่อ admin มี group id 1000 ทำได้โดยทำสั่ง

```
#pico /etc/group
```

จากนั้นเพิ่มบรรทัดต่อไปนี้ที่ท้าย file

admin::1000:

การลบ Group

groupdel <groupname>

#groupdel admin

การเพิ่ม User

ใช้คำสั่ง useradd รูปแบบของคำสั่งมีดังนี้

useradd <Login> -u <User ID> -g <Group> -c FullNamet -d <Home>

การกำหนด password มีรูปแบบคำสั่งดังนี้

passwd <Login>

ตัวอย่างให้เพิ่ม user garfield มี password เป็น usr400

Login = garfield

ค่าอื่นๆที่เหลือ กำหนดตามความเหมาะสมดังนี้

User ID = 1001

Group = users

Full Name = Garfield

Home Directory = /home/garfield

กำหนดให้ Home Directory เป็น /home/garfield directory /home ซึ่งเป็น directory ด้านบน ต้องถูกสร้างก่อน ในกรณีที่ directory ด้านบนของ Home Directory ยังไม่ได้ถูกสร้างต้องทำการสร้างด้วยคำสั่ง

mkdir -p <Parent Directory>

โดยที่ Parent Directory คือ /home (directory ด้านบนของ Home Directory)

ใช้คำสั่ง useradd เพื่อเพิ่ม user โดยมี option ดังนี้

```
#ueradd  garfield -u 1001 -g users -c Garfield -d /home/garfield
```

กำหนดรหัสผ่านเป็น usr400 ด้วยคำสั่ง

```
#passwd  garfield
```


การลบ User

ใช้คำสั่ง userdel โดยรูปแบบคำสั่งดังนี้

userdel -r <username>

โดย option -r จะทำการลบ Home Directory และ Mail Box ของ User

ใช้คำสั่ง userdel เพื่อลบ user garfield

```
#userdel  -r  garfield
```

File และ Directory Permission

คำสั่งที่ใช้แสดงค่า permission ของ file หรือ directory ใช้คำสั่ง

ls -al <ชื่อfileหรือdirectory>

ตัวอย่าง แสดง permission ของ file /etc/passwd ด้วยคำสั่ง ls -al /ect/passwd

ผลลัพธ์ที่ได้

```
#ls  -al  /etc/passwd
-rw-r--r--   1  root     root          876 Jan  4 10:20 /etc/passwd
```

ตัวอย่าง แสดง permission ของ directory /home ด้วยคำสั่ง ls -al /home

ผลลัพธ์ที่ได้

```
#ls  -al   /home
dwxr-xr-x   6  root     root         1024 Nov 20  1998  ./
```

ในกรณีของ directory ให้เลือกเฉพาะรายากรที่มี column สุดท้าย เป็น ./

ส่วน column แรกของผลลัพธ์ (ของ file /etc/passwd) -rw-r--r--มี 10 ตัวอักษร

ตัวแรก - เป็นตัวบอกว่าเป็น file (ถ้าเป็น directory จะเป็นตัว d)

3 ตัวต่อมา rw- เป็นตัวบอก permission ของเจ้าของ file

3 ตัวต่อมา r-- เป็นตัวบอก permission ของ Group

3 ตัวต่อมา r-- เป็นตัวบอก permission ของคนอื่นๆ

ความหมายของตัวอักษรใน permission

file

r สิทธิในการอ่านเนื้อหาภายใน file

w สิทธิในการเขียน/แก้ไขข้อมูลภายใน file

x สิทธิในการเรียก file ขึ้นมาทำงาน

directory

r สิทธิในแสดงรายชื่อ file ภายใน directory

w สิทธิในการเพิ่ม/ลบ/เปลี่ยน ชื่อ file ใน directory

x สิทธิในการเข้าไปใน directory

column ที่สองแสดงเจ้าของ file

column ที่สาม Group ของ file

คำสั่งที่ใช้กำหนด permission

chmod <mode> <ชื่อfileหรือdirectory>

โดยค่าของ mode แทนด้วยตัวเลขสามตัว ตัวแรกแทนสิทธิของเจ้าของ ตัวที่สองแทนสิทธิของ Group ตัวที่สามแทนสิทธิของคนอื่นๆดังนี้

r = 4, w = 2, x = 1, - = 0

ถ้าต้องการเปลี่ยนสิทธิของ file /tmp/test.txt เป็น rxwrw-rw- ส่วนของ mode จะเป็นดังนี้

mode = r+w+x r+w r+w = 4+2+1 4+2 4+2 = 7 6 6

กำหนด permission ด้วยคำสั่ง chmod ตามนี้

```
$ touch /tmp/test.txt
$ ls -l /tmp/test.txt
-rw-r--r--    1 mong     users           0 Jan 14 09:39 /tmp/test.txt
$ chmod 766 /tmp/test.txt
$ ls -l /tmp/test.txt
-rwxrw-rw-    1 mong     users           0 Jan 14 09:39 /tmp/test.txt
```

Compile Software

การติดตั้ง Linux ต้องเลือกกลุ่มของ Package Development และ Development Library ซึ่งมี Compiler และ Library รวมทั้งคำสั่งที่จำเป็นสำหรับการ Compile Software

ตัวอย่างการติดตั้ง Apache Web Server ด้วยการ Compile

ขั้นตอนการ Compile Software

สร้าง Directory /root/src ใช้สำหรับเก็บ Source Code และ Compile

Download Source Code ของ Software นั้นๆ จาก Web Site Download Source Code จาก http://httpd.apache.org โดยมี URL ของ Source Code เป็น

http://www.rge.com/pub/infosystems/apache/httpd/apache_1.3.27.tar.gz

http://www.rge.com/pub/infosystems/apache/httpd/httpd-2.0.43.tar.gz

ถ้ามีคำสั่ง wget สามารถเรียกใช้ได้ดังนี้

wget http://www.rge.com/pub/infosystems/apache/httpd/apache_1.3.27.tar.gz

```
# pwd
/root/src
# ls
apache_1.3.27.tar.gz
```

ได้ File ชื่อ apache_1.3.27.tar.gz

Source Code ส่วนใหญ่ถูกรวมไว้ด้วย tar แล้ว Compress ไว้ด้วย gzip

Uncompress ออกด้วยคำสั่ง

```
#gzip -cd apache_1.3.27.tar.gz | tar xvf -
# ls
apache_1.3.27/  apache_1.3.27.tar.gz
```


ได้ Directory apache_1.3.27

```
# cd apache_1.3.27
# ls
ABOUT_APACHE  Announcement  cgi-bin/  conf/  config.layout  configure*
htdocs/  icons/  INSTALL  LICENSE  logs/  Makefile.tmpl  README
README.configure  README-WIN.TXT  src/  WARNING-WIN.TXT
```

สำหรับการติดตั้งแบบธรรมดา (รายละเอียดอยู่ใน File INSTALL)

ตรวจสอบส่วนสภาพแวดล้อมของระบบและเป็นสิ่งจำเป็นสำหรับการ Compile ด้วยคำสั่ง configure Apache จะติดตั้งไว้ที่ Directory /usr/local/apache ถ้าต้องการเปลี่ยน Directory ที่ติดตั้งให้เพิ่ม Option --prefix=<Directory Name>

./configure --prefix=/usr/webserver

ติดตั้ง Apache ไว้ที่ Directory /usr/webserver

หลังจาก configure เรียบร้อยแล้วเรียกใช้คำสั่ง make เพื่อ Compile Source Code แล้วตามด้วย make install เพื่อติดตั้ง Apache ไว้ตาม Directory ที่กำหนด

```
# ./configure
# make
# make install
# /usr/local/apache/bin/apachectl start
```

เพิ่ม /usr/local/apache/bin/apachectl ลงใน file /etc/rc.d/rc3.d/S90apache เพื่อให้ Apache Start ทุกครั้งที่ Start เครื่อง Server และเพิ่มลงใน file /etc/rc.d/rc3.d/K90apache เพื่อให้ Apache Start ทุกครั้งที่ Shutdown เครื่อง Server

rpm (RedHat Package Manager)

เป็นโปรแกรมจัดการ (ติดตั้ง ยกเลิกการติดตั้ง เพิ่มเติม สร้าง) Package
Package คือชุดของโปรแกรมซึ่งประกอบไปด้วย file ต่างๆที่จำเป็นสำหรับการทำงานของโปรแกรมนั้นๆ ซึ่งก่อนที่จะมาเป็น Package โปรแกรมส่วนใหญ่ที่มีอยู่บน Linux จะอยู่ในรูปของ Source Code ซึ่งก่อนจะนำมาใช้งานได้นั้นจำเป็นต้อง Compile และ Link ส่วนที่จำเป็นต่างๆเข้าด้วยกัน จนได้เป็นโปรแกรมที่พร้อมทำงานออกมา ซึ่งขั้นตอนต่างๆนั้นค่อนข้างยุ่งยากและมักมีปัญหาอยู่บ่อยๆ ดังนั้นจึงได้มีการจัดนำ file ต่างๆที่ได้จากการ Compile และ Linkเรียบร้อยแล้ว รวมทั้งส่วนต่างๆที่จำเป็น นำมารวมไว้เป็น Package จากนั้นจึงนำแจกจ่ายออกไป
ผู้ที่ต้องการติดตั้งโปรแกรมจึงไม่จำเป็นต้องทำการ Compile Source Code อีกต่อไป หน้าที่ของผู้ติดตั้งที่ต้องทำก็คือนำ Package File ที่ได้มา copy ไว้ในตำแหน่งที่ถูกต้องบนเครื่องเท่านั้น
โปรแกรมที่ช่วยจัดการเกี่ยวกับ Package ก็คือ โปรแกรม rpm
Package File ของ RedHat จะเก็บอยู่ในแผ่น CDROM ที่ใช้ติดตั้ง โดยเก็บอยู่ใน directory /RedHat/RPMS ดังนั้นถ้าต้องการติดตั้งโปแกรมเพิ่มเติมทำได้โดยการ mount CDROM แล้วติดตั้งโดยใช้ file ใน directory ดังกล่าว

ชื่อของ Package File แบ่งออกเป็น 4 ส่วนโดยมีรูปแบบดังนี้
Name-Version-Release.Architecture.rpm
1 Name คือชื่อของ Package จะตั้งชื่อตามโปรแกรมที่อยู่ใน Package
2 Version คือเลข version ของโปแกรม ที่อยู่ใน Package
3 Release คือเลข release ของ Package เป็นตัวบอกว่า Package สร้างเป็นครั้งที่เท่าไร
4 Architecture คือ สถาปัตยกรรมของ CPU เช่น i386 หมายถึง CPU ตระกูล Intel และที่ Compatible กับ Intel เช่น AMD , Cyrix
ถ้าเป็น noarch แสดงดว่าไม่ขึ้นอยู่กับสถาปัตยกรรมคือใช้ได้ทุกๆประเภทของ CPU

ตัวอย่างชื่อ Package File ของ Apache Web Server
apache-1.3.6-7.i386.rpm

ส่วนที่สำคัญก่อนการติดตั้งโปรแกรมจาก Package คือ
1 Architecture จะต้องเลือกให้ตรงกับเครื่องที่ใช้
2 ต้องเลือกให้ตรงกับ version ของ RedHat ที่ใช้ เช่นติดตั้ง RedHat version 6.0 ก็ควรหา Package ที่สร้างมาสำหรับ version 6.0 ถ้าใช้ไม่ตรงกัน อาจทำให้ไม่ใช้งานได้ในบาง Package

การใช้งานคำสั่ง rpm เบื้องต้น
คำสั่งสำหรับแสดงรายละเอียดของ Package ที่ถูกติดตั้งไว้แล้วในเครื่อง มีรูปแบบคำสั่งดังนี้
rpm -qi <PackageName>

ตัวอย่างเช่น ถ้าต้องการรู้ว่าในเครื่องติดตั้ง Package ของ Apache Web Server ไว้แล้วหรือยัง ทำได้โดยใช้คำสั่งดังนี้ (ชื่อ Package ของ Apache Web Server คือ apache)

```
#rpm  -qi  apache
```

ถ้ามีการติดตั้งแล้วจะแสดงรายการดังนี้

```
Name        : apache                       Relocations: (not relocateable)
Version     : 1.3.12                            Vendor: Red Hat, Inc.
Release     : 25                            Build Date: Thu 24 Aug 2000
02:45:27 AM ICT
Install date: Sun 21 Oct 2001 04:50:53 AM ICT      Build Host:
porky.devel.redhat.com
Group       : System Environment/Daemons    Source RPM: apache-1.3.12-
25.src.rpm
Size        : 1179253                          License: Freely distributable
.
```

โดยในรายการจะบอกถึงวันเวลาที่ติดตั้ง รวมทั้งรายละเอียดต่างๆ

ถ้ายังไม่ได้ติดตั้งจะได้ผลลัพธ์ดังนี้

```
package apache is not installed
```

คำสั่งแสดงชื่อ file และ directory ต่างๆ ใน Package ที่ถูกติดตั้งไว้แล้วในเครื่องมีรูปแบบคำสั่งดังนี้

rpm -ql <PackageName>

ตัวอย่างเช่น ถ้าต้องการรู้ว่ามี file ใดบ้างใน Package ของ Apache Web Server ที่ถูกติดตั้งไว้ในเครื่อง ทำได้โดยใช้คำสั่งดังนี้

```
#rpm -ql  apache
```

ถ้ามี Apache Web Server ติดตั้งอยู่จะมีรายชื่อของ file และ directory แสดงดังนี้

```
/etc/httpd/conf
/etc/httpd/conf/access.conf
/etc/httpd/conf/httpd.conf
        …..more…..
```

คำสั่งดูรายละเอียดของ Package File มีรูปแบบคำสั่งดังนี้

rpm –qpi <PackageFile>

คำสั่งดูชื่อ file และ directory ต่างๆ ที่อยู่ใน Package File มีรูปแบบคำสั่งดังนี้

rpm –qpl <PackageFile>

คำสั่งติดตั้ง Package มีรูปแบบคำสั่งดังนี้

rpm –i <PackageFile>

กำหนดให้ CDROM mount อยู่ที่ /mnt

Package File ทั้งหมดเก็บอยู่ใน directory /mnt/RedHat/RPMS

เข้าไปใน directory /mnt/RedHat/RPMS ด้วยคำสั่ง

```
#cd /mnt/RedHat/RPMS
```

ตัวอย่างเช่นถ้าต้องการติดตั้ง Proxy Server โดยมีชื่อ Package ว่า squid

โดยทั่วไปชื่อ file ของ Package จะขึ้นต้นด้วยชื่อของ Package

ดังนั้นจึงสามารถอ้างอิงถึงชื่อ file ของ Package Proxy Server ได้ดังนี้

squid*

ดูรายละเอียดของ Package File โดยใช้คำสั่ง

```
#rpm  -qpi  squid*
```

ดูชื่อ file และ directory ต่างๆใน Package File โดยใช้คำสั่ง

```
#rpm   -qpl  squid*
```

ติดตั้ง Package โดยใช้คำสั่ง

```
#rpm –i  squid*
```

คำสั่งยกเลิกการติดตั้ง Package มีรูปแบบคำสั่งดังนี้

rpm –e <PackageName>

ตัวอย่างเช่น ถ้าต้องการยกเลิกการติดตั้ง Proxy Server โดยมีชื่อ Package ว่า squid ทำได้โดยใช้คำสั่งต่อไปนี้ (ก่อนยกเลิกการติดตั้งควรสั่งให้ Proxy Server จบการทำงานก่อน)

```
#rpm -e squid
```

การติดตั้ง package ผ่าน ftp Server

rpm -i ftp://<user>:<password>@hostname:<port>/path/to/package.rpm

ตัวอย่าง

```
#rpm -i ftp://ftp.rpmfind.net/linux/Mandrake-devel/cooker/RPMS/squid-2.5.STABLE1-3mdk.rpm
```

การ rebuild package

Download Source squid-2.5.STABLE1-3mdk.src.rpm จาก http://www.rpmfind.net/

```
#rpm --rebuild squid-2.5.STABLE1-3mdk.rpm
```

เมื่อ rebuild เรียบร้อยแล้วจะได้ Package (rpm) เก็บไว้ใน Directory /usr/src/RPM/RPMS/ (สังเกตจากผลลัพธ์ที่ได้) จากนั้นใช้คำสั่ง rpm -i เพื่อทำการติดตั้ง

รายละเอียดเพิ่มเติม อ่านได้จากคำสั่ง man rpm

การจัดการ Service

คำสั่ง service ใช้สำหรับแสดงสถานะการทำงานของ service ต่างๆในระบบ และใช้ควบคุมการทำงานของ service เช่นสั่งให้ service เริ่มต้นทำงาน (start) สิ้นสุดการทำงาน (stop) เริ่มทำงานใหม่ (restart) ซึ่งกรณีของ restart จะใช้เมื่อมีการเปลี่ยนค่าเริ่มต้นการทำงานของ service นั้น ปกติแล้วการเปลี่ยนค่าเริ่มต้นการทำงานจะไม่มีผลต่อ service จนกว่าจะสั่ง restart service

รูปแบบของคำสั่ง

service < option > | --status-all | [service_name [command | --full-restart]]

การแสดงสถานะการทำงานของ service ทำได้โดยใช้ option --status-all

```
# service --status-all
anacron is stopped
apmd is stopped
atd (pid 359) is running...
crond (pid 558) is running...
dhcpd is stopped
gpm is stopped
httpd (pid 2151 2150 2149 2148 2147 2146 2145 2144 543) is running...
```

รูปแบบคำสั่งการควบคุม service

service service_name start|stop|restart

ตัวอย่างเช่นต้องการ stop service httpd ใช้คำสั่งดังนี้

```
# service httpd stop
Shutting down http:                                        [  OK  ]
```

ถ้าต้องการ start service httpd ใช้คำสั่งดังนี้

```
# service httpd start
Starting httpd:                                            [  OK  ]
```

ถ้าต้องการ restart service httpd ใช้คำสั่งดังนี้

```
# service httpd restart
Shutting down http:                                        [  OK  ]
Starting httpd:                                            [  OK  ]
```

สถานะของ Service เมื่อเครื่อง Boot

การกำหนดสถานะการทำงานของ Service เมื่อเครื่อง Boot ด้วยคำสั่ง cfgconfig มีรูปแบบของคำสั่งดังนี้

```
# chkconfig
chkconfig version 1.3.4 - Copyright (C) 1997-2000 Red Hat, Inc.
This may be freely redistributed under the terms of the GNU Public
License.

usage:   chkconfig --list [name]
         chkconfig --add <name>
         chkconfig --del <name>
```

แสดงสถานะการทำงานของ Service ต่างๆ ด้วยคำสั่ง chkconfig --list

```
# chkconfig --list
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
partmon         0:off   1:off   2:off   3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
ipsec           0:off   1:off   2:off   3:off   4:off   5:off   6:off
webmin          0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd based services:
        chargen-udp:    off
        chargen:        off
        daytime-udp:    off
        daytime:        off
        echo-udp:       off
        echo:   off
        fam:    on
        rsync:  off
        cvs:    off
        proftpd-xinetd: off
```

ตรวจสอบสถานะของ proftpd-xinetd แล้วกำหนดให้ทำงานทุกครั้งที่เครื่อง Boot ด้วยคำสั่ง

chkconfig --add proftpd-xinetd

```
#chkconfig --list proftpd-xinetd
proftpd-xinetd  off

#chkconfig --add proftpd-xinetd

#chkconfig --list proftpd-xinetd
proftpd-xinetd  on
```


ตรวจสอบสถานะของ webmin ถ้าต้องการกำหนดให้ webmin ไม่ทำงานเมื่อเครื่อง Boot ใช้คำสั่ง

chkconfig --del webmin

```
#chkconfig --list webmin
webmin          0:off   1:off   2:on    3:on    4:on    5:on    6:off

#chkconfig --del webmin

#chkconfig --list webmin
webmin          0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

xinetd (extended Internet services daemon)

การทำงานของ Service ใน linux มีการทำงานแบ่งเป็น 2 ประเภท
1 ทำงานแบบ daemon จะเริ่มต้นทำงานตอนเปิดเครื่องและทำงานอยู่ตลอดเวลา
2 ทำงานภายใต้การควบคุมของ xinetd จะลงทะเบียนไว้กับ xinetd เวลาที่มีผู้ใช้งานต้องการติดต่อ xinetd จะเป็นคนเรียก Service ที่ลงทะเบียนไว้ขึ้นมาทำงาน (ตามที่ผู้ใช้งานร้องขอ) เมื่อให้บริการเรียบร้อยแล้ว Service ก็จะจบการทำงานลง

การทำงานแบบ daemon จะใช้สำหรับ Service ที่มีการใช้งานบ่อยๆเพื่อประสิทธิภาพการให้บริการ
ส่วนการทำงานภายใต้ xinetd จะใช้กับ Service ที่ทำงานไม่บ่อย เพื่อประหยัดทรัพยากรของระบบ
xinetd เป็น Service (daemon) ที่ทำหน้าที่เรียก Service อื่นๆที่ลงทะเบียนไว้กับ xinetd ขึ้นมาทำงาน
file ที่เก็บรายละเอียดของ Service ที่ลงทะเบียนไว้กับ xinetd คือ /etc/xinetd.conf และ file ทั้งหมดที่อยู่ใน directory /etc/xinetd.d
ทุกครั้งที่มีการเปลี่ยนแปลงค่าของ file ใน directory /etc/xinetd.d จะต้องสั่ง restart Service xinetd ทุกครั้งเพื่อให้ xinetd รู้ถึงการเปลี่ยนแปลงและทำงานตรงกับค่าที่ทำการแก้ไข
รูปแบบคำสั่ง

```
#service  xinetd  restart
```

เมื่อมีการติดตั้ง Service จาก Package โดยใช้คำสั่ง rpm ถ้า Service นั้นมีการทำงานแบบ xinetd จะมีการเพิ่ม file ลงใน directory /etc/xinetd.d ให้อัตโนมัติ
ถ้าต้องการให้ Service ที่ติดตั้งทำงานได้ภายใต้ xinetd ต้องเข้าไปตรวจสอบสถานะการทำงานภายใต้ xinetd โดยใช้คำสั่ง setup แล้วเลือก System services แล้วเลือกรายการของ Service ที่ได้ติดตั้งลงไปให้มีเครื่องหมาย * นำหน้า แล้วสั่งให้ xinetd restart
หมายเหตุ ถ้าเป็น Linux Version เก่า หรือ UNIX จะใช้ inetd และ file ที่เก็บ Service ที่ลงทะเบียนไว้คือ /etc/inetd.conf

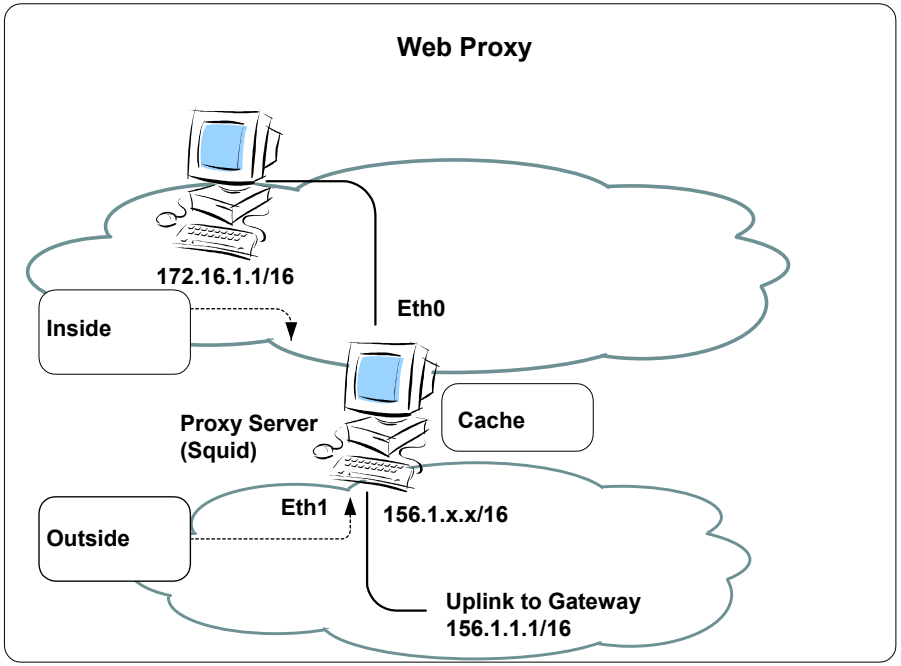
file /etc/xinetd.conf

```
#cd  /etc/xinetd.d
#cat imap

# default: off

service imap
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/imapd
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        disable                 = yes
}
```

Web Proxy (Squid)

http://www.squid-cache.org/

รูปแบบการทำงาน



ค่าเริ่มต้นการทำงานเก็บอยู่ใน file /etc/squid/squid.conf

การกำหนดค่าเริ่มต้นการทำงานของ squid

กำหนดรายละเอียดการให้บริการใน file /etc/squid/squid.conf

```
#pico  /etc/squid/squid.conf
```

ค่าต่างๆที่กำหนดใน file /etc/squid/squid.conf มีรายละเอียดดังนี้

http_port 3128

กำหนดหมายเลข port ที่ให้บริการสำหรับ Web Browser เป็น 3128

cache_mem 8 MB

กำหนดขนาดของ memory ที่ใช้เก็บ cache เป็น 8 MB

cache_dir ufs /var/spool/squid 100 16 256

กำหนด ระบบ file ที่เก็บ cache,ตำแหน่งของ directory หลักที่เก็บ cache ,ขนาดเนื้อที่ disk ที่เก็บ cache (หน่วยเป็น MB),จำนวน directory ย่อยลำดับที่ 1,จำนวน directory ย่อยลำดับที่ 2 ตามลำดับ

cachemgr_passwd secret all

กำหนดรหัสผ่านสำหรับเข้าจัดการ Proxy Server ผ่าน Web

http://servername/cgi-bin/cachemgr.cgi

copy file cachemgr.cgi ไปไว้ภายใต้ Directory การทำงานของ Web Server ด้วยคำสั่ง

cp /usr/lib/squid/cachemgr.cgi /var/www/cgi-bin/cachemgr.cgi

การควบคุมการเข้าใช้บริการ (Access Control)

การกำหนดชื่อแทนกลุ่มเครื่อง

รูปแบบ

```
acl    <acl name>    src         <IP/Netmask>
acl    <acl name>    dst         <IP/Netmask>
acl    <acl name>    srcdomain   <DomainName>
acl    <acl name>    dstdomain   <DomainName>
```

src	กลุ่มเครื่องที่ติดต่อเข้าหา Proxy Server
dst	กลุ่มเครื่องที่ Proxy Server ติดต่อไปหา
srcdoamin	ชื่อ Domain ที่ติดต่อเข้าหา Proxy Server
dstdomain	ชื่อ Domain ที่ Proxy Server ติดต่อไปหา
<acl name>	เป็นชื่อแทนกลุ่มของเครื่อง
<IP/Netmask>	เป็น IP Address และ Netmask ของเครื่องในกลุ่ม
<DomianName>	เป็นชื่อ Domain ของเครื่องในกลุ่ม

การกำหนดสิทธิ์การใช้งาน

รูปแบบ

```
http_access  allow  <acl name>
http_access  deny   <acl name>
```

ตัวอย่าง

```
acl    all    src    0.0.0.0/0.0.0.0
```


เป็นการกำหนดชื่อกลุ่มของผู้ที่เข้ามาใช้บริการโดยให้ชื่อว่า all แทน เครื่องที่มาจาก IP Address ใดๆก็ได้

```
http_acccess    allow   all
```

เป็นการกำหนดสิทธิในการเข้าใช้งานตัว Proxy Server โดยให้เครื่องที่อยู่ในกลุ่ม all สามารถเข้าใช้งานได้

ทดลองเปลี่ยน ส่วน Access Control ใน file /etc/squid/squid.conf

```
acl    intranet    src      172.16.0.0/255.255.0.0
acl    thai        dstdomain    .th
acl    com         dstdomain    .com

http_access  allow  intranet
http_access  deny   com
http_access  allow  thai
http_access  deny   all
```

สร้าง directory ย่อยสำหรับเก็บข้อมูล cache ด้วยคำสั่ง (ทำครั้งเดียวหลังจากติดตั้งโปรแกรม)

```
#squid  -z
```

เริ่มต้นให้บริการด้วยคำสั่ง

```
#service squid start
```

เมื่อมีการแก้ไขค่าใน file /etc/squid/squid.conf ต้องสั่งให้ squid restart ทุกครั้ง ด้วยคำสั่ง

```
#service squid restart
```

ทดสอบการทำงานเบื้องต้นด้วยการ telnet แล้วตามด้วยหมายเลข port ของ Squid (ค่า Default port ของ squid คือ 3128) จะได้ผลลัพธ์

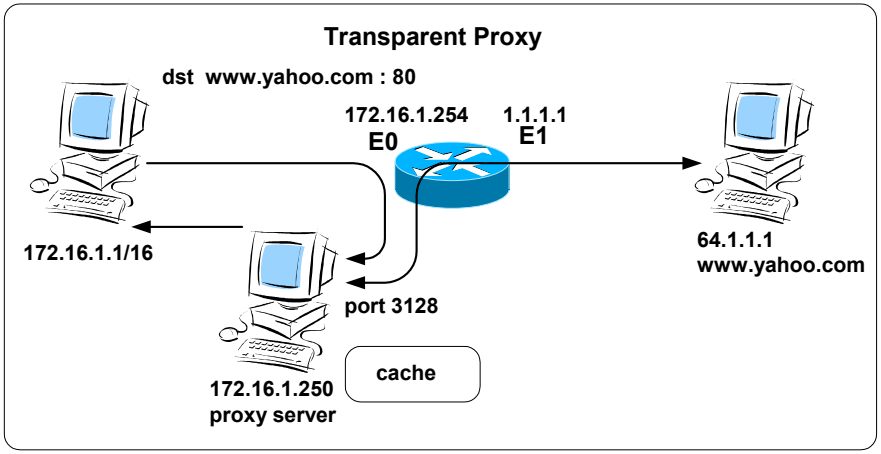
```
# telnet localhost 3128
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

แสดงว่ามี Service ทำงานที่ port 3128

กำหนดให้เริ่มต้นให้บริการทุกครั้งที่เครื่อง boot ด้วยคำสั่ง

```
#chkconfig --add squid
```


Transparent Proxy



การใช้งาน Web Proxy ที่ Web Browser ของเครื่อง Client ต้องกำหนดชื่อเครื่อง (IP Address) และหมายเลข Port ของ Proxy Server เพื่อติดต่อกับ proxy server ให้เป็นตัวแทนไปนำข้อมูลจาก Web Server

ค่าเริ่มต้นของ TCP/IP ที่จำเป็นสำหรับเครื่อง Client คือ IP Address Subnet Mask โดยไม่จำเป็นต้องมี Gateway ถ้าเครื่องที่เป็น Proxy Server และเครื่อง Client อยู่ใน Network เดียวกัน (เพราะสามารถติดต่อกันได้โดยตรง) และไม่จำเป็นต้องกำหนดค่า DNS Server ให้กับเครื่อง Client เพราะการ Resolve ค่าของ Hostname เป็นหน้าที่ของเครื่อง Proxy Server

แต่ถ้าจำนวนเครื่อง Client มีจำนวนมาก การกำหนดค่าของให้กับ Browser เพื่อติดต่อมายัง proxy server จะทำให้ไม่สามารถทำได้ง่ายนัก และถ้าเครื่อง Client ใดไม่ได้กำหนดค่า proxy server จะไม่สามารถเข้าใช้งานเว็บได้ หรือเข้าใช้งานได้โดยไม่ผ่าน proxy server ถ้าเครื่อง Client นั้นต่อโดยตรงกับ internet

การใช้งาน Transparent Proxy ใช้หลักการของการ Redirect TCP Port โดยถ้าข้อมูลที่ส่งจากเครื่อง Client เพื่อติดต่อไปยัง Web Server Destination Port จะเป็น Port ของ www (80) ที่ Gateway จะทำการ Redirect Packet นั้นไปยัง port ของ Proxy Server (3128) จากนั้น Proxy Server จะติดต่อกับ Web Server และนำข้อมูลมาให้ Client

การใช้ Linux และ Squid ทำงานเป็น Transparent Proxy

กำหนดค่าเริ่มต้นใน file /etc/squid/squid.conf ในส่วนของ HTTPD-ACCELERATOR OPTIONS

Option เดิมจะถูก comment ไว้ด้วย #

```
#httpd_accel_host hostname
#httpd_accel_port port
#httpd_accel_with_proxy off
#httpd_accel_uses_host_header off
```

ให้เปลี่ยนเป็น

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

ที่ iptables กำหนดให้ redirect packet ที่ส่งจาก เครื่อง client ไปยัง port ปลายทางหมายเลข 80 (port ของ www) ไปยัง port 3128 (port ของ squid) โดยกำหนดที่ nat table

```
#iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

ถ้าใช้ ipchains (สำหรับ Kernel 2.2) กำหนดที่ input chain ด้วยคำสั่ง

```
ipchains -A input -p  tcp –s 0.0.0.0/0 -d 0.0.0.0 80 -j REDIRECT 3128
```


Router และ Firewall โดยใช้ Linux

จุดประสงค์การเรียนรู้

สามารถติดตั้งและกำหนดค่าเริ่มต้นของ Network Card หลาย Card บนเครื่อง Linux Server

สามารถกำหนด Access Control ในระดับ Service ของ TCP Wrapper ได้

สามารถอธิบายหลักการทำงาน ส่วนประกอบของ iptables ในส่วนของ Chain และสามารถเขียน Rule และกำหนด Action เพื่อทำงานกับ Packet ได้

สามารถใช้งาน table Filter และ table NAT ใน iptables ได้

สามารถใช้งาน iptables ทำงานร่วมกันกับ Transparence Proxy ได้

สามารถติดตั้งและใช้งาน Shorewall เพื่อทำงานเป็น Firewall แบบ Single Interface, Two Interface

สามารถใช้งาน Shorewall แบบ SNAT, Redirect, DNAT, Masquerade ได้

สามารถใช้งาน Shorewall แบบ IPSec Tunnel ทำงานร่วมกับ IPSec (Freeswan) ได้

Website อ้างอิง

http://www.iptables.org/

http://www.shorewall.net/

ติดตั้ง Package shorewall

shorewall-1.3.7c-1mdk.noarch.rpm

การ Config Linux Server ให้ทำหน้าที่เป็น Router

Network Configuration

การใช้งาน Network Card และการกำหนดค่าเริ่มต้น TCP/IP

File /etc/modules.conf เก็บรายละเอียดของการเรียกใช้งาน modules ต่างๆ ของระบบ

คำสั่งสำหรับการกำหนดค่า module, irq, io address ของ Network Card

```
alias <device name> <module name>
```

device name คือชื่อเรียกของ Network Card ที่ใช้ตอนกำหนดค่า IP Address (จากคำสั่ง netconf)

module name เป็นชื่อ module ของ Network Card

Module ของ Network Card เก็บไว้ใน Directory /lib/modules/2.4.19-16mdk/kernel/drivers/net/ (2.4.19-16mdk เป็น Version ของ Kernel ซึ่งอาจต่างกันไปตาม Distribution)

NE2000 Compatible เป็นมาตรฐานของ Network Card ส่วนใหญ่สนับสนุน ดังนั้นถ้าหา module ที่ตรงกับ Card ไม่ได้ให้เลือกใช้เป็น ne

ในกรณีที่เป็น Network Card แบบ ISA การใช้ module แบบ ne ต้องกำหนด irq และ io address ของ Card มีรูปแบบเป็นตัวเลขฐาน 16 (เขียนโดยเริ่มต้นด้วย 0x)

ในกรณีที่มี Network Card ที่ใช้ Module ne เหมือนกันหลาย Card การกำหนด irq และ io address ให้ขั้นแต่ละชุดด้วย comma (,)

```
options <module name> <option>
```


ตัวอย่าง Network Card แบบ ISA 2 Card เลือก Module แบบ ne โดย Card แรก มี io address เป็น 0x280 irq เป็น 0x9 และ Card ที่สองมี io address เป็น 0x320 และ irq เป็น 0x5 สามารถเขียน file /etc/modules.conf ได้ดังนี้

```
alias eth0 ne
alias eth1 ne
options ne io=0x280,0x320 irq=0x9,0x5
```

(รายละเอียดของ module อื่นๆ ที่ไม่เกี่ยวข้องกับ Network Card ให้เก็บไว้เหมือนเดิม)

หลังจากกำหนดรายละเอียดใน file /etc/modules.conf เรียบร้อยแล้วจึงเรียกใช้งาน module โดยมีคำสั่งที่เกี่ยวข้องกับ module

modprobe	<module name>	เรียกใช้งาน module
rmmod	<module name>	ยกเลิก module ที่ใช้งานอยู่
lsmod		แสดง module ที่เรียบใช้อยู่

ตัวอย่างคำสั่งจัดการ Kernel Module

```
#rmmod  ne
#modprobe  ne
#lsmod
```

ในกรณีที่ Network Card ถูกใช้งานอยู่แล้วต้องการยกเลิกการใช้งาน module จะทำไม่ได้ในทันที ต้องยกเลิกการใช้งาน Network Card นั้นก่อน โดยใช้คำสั่ง service network stop

กำหนดค่า IP Address และค่าที่เกี่ยวข้องให้กับ Network Card

เรียกใช้คำสั่ง netconf แล้วเลือก Host name and IP network devices

กำหนด IP Address และค่าอื่นๆ ตามรายละเอียดของ TCP/IP

สำหรับ Network Card 1

```
Net device      eth0
Kernel module   ne
I/O port (opt)        0x280,0x320
Irq (opt)       0x9,0x5
```

สำหรับ Network Card 2

```
Net device      eth1
Kernel module   ne
I/O port (opt)
Irq (opt)
```

ออกจาก netconf แล้วสั่ง restart network ด้วยคำสั่ง

```
#service network restart
```


ตรวจสอบค่า IP ด้วยคำสั่ง ifconfig

```
#ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:C8:7D:73:5B
          inet addr:170.16.1.252  Bcast:170.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:406980 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32394 errors:0 dropped:0 overruns:0 carrier:0
          collisions:5 txqueuelen:100
          RX bytes:37575478 (35.8 Mb)  TX bytes:4743935 (4.5 Mb)
          Interrupt:9 Base address:0x280

eth1      Link encap:Ethernet  HWaddr 00:00:E8:1B:78:D5
          inet addr:195.168.3.254  Bcast:195.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:406073 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31777 errors:3 dropped:0 overruns:0 carrier:3
          collisions:76 txqueuelen:100
          RX bytes:35912653 (34.2 Mb)  TX bytes:3104290 (2.9 Mb)
          Interrupt:5 Base address:0x320
```

IP Forwarding

เครื่องที่ทำหน้าที่เป็น Router ต้องสามารถ Forward IP ได้

ตรวจสอบ function การทำงานของ ip forward จากคำสั่ง sysctl ต้องได้ผลลัพธ์เป็น 1

```
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
```

ถ้ามีค่าเป็น 0 ให้ใช้คำสั่ง sysctl -w net.ipv4.ip_forward=1 และเข้าไปแก้ไขค่าใน file /etc/sysctl.conf

```
# sysctl -w net.ipv4.ip_forward=1
```

```
# pico /etc/sysctl.conf
```

ภายใน file ประกอบด้วย option ต่างๆดังนี้

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Disables automatic defragmentation (needed for masquerading, LVS)
net.ipv4.ip_always_defrag = 0
# Disables the magic-sysrq key
kernel.sysrq = 0
```

กำหนด option net.ipv4.ip_forward ให้มีค่าเป็น 1

net.ipv4.ip_forward = 1

การเปลี่ยนแปลงจะมีผลเมื่อ boot เครื่องครั้งต่อไป

กำหนด Default Route เป็น Gateway ที่ต่ออยู่กับ Internet

Service บน Router และ Firewall

เครื่องที่ทำหน้าที่เป็น Firewall ควรมี Service เฉพาะที่จำเป็นเท่านั้น Service ใดไม่จำเป็นให้ปิดไป (เรียกใช้คำสั่ง linuxconf แล้วเลือก Control panel -> Control service activity) ส่วน Service ที่ใช้งานอยู่ ควรควบคุมให้เฉพาะ Client (IP Address) ที่เกี่ยวข้องเท่านั้นที่สามารถเข้ามาใช้งานได้

TCP Wrapper

เป็นการควบคุม Service ที่ทำงานอยู่ภายใต้การควบคุมของ xinetd (inetd ใน UNIX หรือ Linux version ก่อนๆ) ให้สามารถเรียกใช้งานได้ (หรือห้ามใช้งาน) จาก Client ที่ระบุไว้เท่านั้น เป็นการทำ Access Control ในระดับของ Service

File ที่เกี่ยวของกับการทำงาน /etc/hosts.allow และ /etc/hosts.deny

รูปแบบของ Access Control

```
<Service Name>:<Client>
```

Service เป็นชื่อของ Service ที่อยุ่ภายใต้การควบคุมของ xinetd (รายละเอียดอยู่ใน directory /etc/xinetd.d) เช่น telnet, finger, talk

Client IP Address ของเครื่อง Cleint

ALL ใช้แทน Service, Client ทั้งหมด

/etc/hosts.allow

เก็บรายละเอียดของ Client ที่สามารถใช้งานได้

```
ALL:192.168.1.1
telnet:192.168.1.0/255.255.255.0
```

อนุญาตให้ Client ที่มี IP Address 192.168.1.1 สามารถเรียกใช้งาน Service (ภายใต้การควบคุมของ xinetd) ได้ทุก Service
เครื่องจาก Subnet 192.168.1.0/255.255.255.0 สามารถใช้งาน Service telnet ได้

/etc/hosts.deny
เก็บรายละเอียดของ Client ที่ห้ามเข้าใช้งาน

```
ALL: 0.0.0.0/0.0.0.0
```

Client ที่ไม่ตรงกับเงื่อนไขใน file /etc/hosts.allow ถูกห้ามเข้าใช้งาน Service ทุก Service

(ดูรายละเอียดจากคำสั่ง man hosts.allow และ man hosts.deny)

iptables

iptables ประกอบด้วย 3 table คือ filter,nat,mangle โดยใน table filter ประกอบด้วย 3 Chain คือ

INPUT Packet ที่มีปลายทางมาที่เครื่อง Server
FORWARD Packet ที่ส่งผ่านมาที่เครื่อง Server เพื่อส่งต่อไปยังปลายทางที่อื่น
OUTPUT Packet ที่ถูกส่งออกจากเครื่อง Server เอง

ที่ Command Line เรียกใช้คำสั่ง iptables -L ได้ผลลัพธ์ดังนี้

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
     target     prot opt source          destination
```

ในแต่ละ Chain ประกอบไปด้วย rule (เงื่อนไขการทำงาน)
การจัดการ rule มี Option ดังนี้

-A Append rule
-I Insert rule
-R Replace rule
-D Delete rule

ในแต่ละ Chain มี policy (ถูกใช้ในกรณีที่ Packet ที่ตรวจสอบไม่ตรงตามเงื่อนไขที่เขียนไว้ Packet นั้นจะถูกจัดการตามค่าที่กำหนดไว้ใน policy ค่าของ policy คือ ACCEPT หรือ DROP)

Option สำหรับกำหนดเงื่อนไขการทำงานของแต่ละ rule
-j กำหนด target

ACCEPT	Packet ที่ถูก Accept ไม่ต้องผ่าน rule ที่เหลือใน Chain
DROP	Packet จะถูก Drop ทิ้ง
REJECT	เหมือนกันกับ DROP แต่จะมีการส่ง error กลับไปบอกผู้ส่ง

Option สำหรับกำหนดเงื่อนไขอ้างอิงถึง packet

-i	input interface
-o	output interface
-p	protocol
-s	source ip address
-d	destination ip address
--sport	source port
--dport	destination port
--mac-source	source mac address

รูปแบบคำสั่งการเพิ่ม rule

```
iptables -A  <ชื่อ Chain>  <เงื่อนไขของ Packet>  -j  <target>
```

ตัวอย่างคำสั่ง

```
iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
```

Packet จาก FORWARD Chain ที่ติดต่อออกไปยัง Web Server (port 80) ถูก forward ออกไปได้


```
iptables -A INPUT -p icmp -j DROP
```

Packet icmp (ping) จาก INPUT Chain จะถูก drop

```
iptables -A INPUT -s 192.168.0.0 -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -s ! 192.168.0.0 -p tcp --dport ssh -j DROP
iptables -A OUTPUT -p tcp --dport telnet -j DROP
```

Packet จาก INPUT Chain ที่มาจาก ip address 192.168.0.0 มี Protocol เป็น tcp มี Port ปลายทางไปที่ Service ของ ssh (Port 22) จะถูกส่งผ่านไปได้ ส่วน Packet จาก INPUT Chain ในเงื่อนไขเดียวกันแต่ไม่ได้มาจาก ip address 192.168.0.0 จะถูก Drop
Packet จาก Output Chain ที่ติดต่อไปยัง Service ของ telnet (Port 23) จะถูก Drop

```
iptables -A INPUT --mac-source 00:00:00:00:00:01 -j DROP
```

Packet จาก INPUT Chain ถ้ามี MAC Address เป็น 00:00:00:00:00:01 จะถูก Drop ทิ้ง

Option --state

ใน TCP มีขั้นตอนเริ่มต้นการติดต่อเรียกว่า three-way handshake การติดต่อจาก Client ไปยัง Server มีขั้นตอนดังนี้

```
        Client          Server

1       SYN  --->
2                <--- SYN+ACK
3       ACK  --->
4                <--- ACK
5       ACK  --->


9       FIN+ACK --->
10               <--- ACK
11               <---FIN+ACK
12      ACK  --->
```

1 Client ส่ง SYN ไปให้ Server เพื่อขอสร้างการติดต่อจาก Client -> Server

2 Server ส่ง ACK เป็นการบอก Client ว่าการติดต่อจาก Client -> Server ที่ขอเรียบร้อย พร้อมกับส่ง SYN เพื่อขอสร้างการติดต่อจาก Server -> Client

3 Client ส่ง ACK เป็นการบอก Server ว่าการติดต่อที่ขอเรียบร้อย

4 เป็นต้นไปเป็นการรับส่งข้อมูลกันระหว่าง Client และ Server

INVALID	แทน packet ที่ไม่เกี่ยวข้องใดกับการติดต่อนั้นๆ
ESTABLISHED	แทน packet ที่สัมพันธ์กันกับการติดต่อที่เกิดขึ้นทั้งสองทิศทาง
NEW	แทน packet ที่ขอเปิดการติดต่อขึ้นมาใหม่
RELATED	แทน packet ที่ขอเปิดการติดต่อขึ้นมาใหม่ โดยมีความสัมพันธ์ (เกิดขึ้นเนื่องจาก) กับการติดต่อที่มีอยู่

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

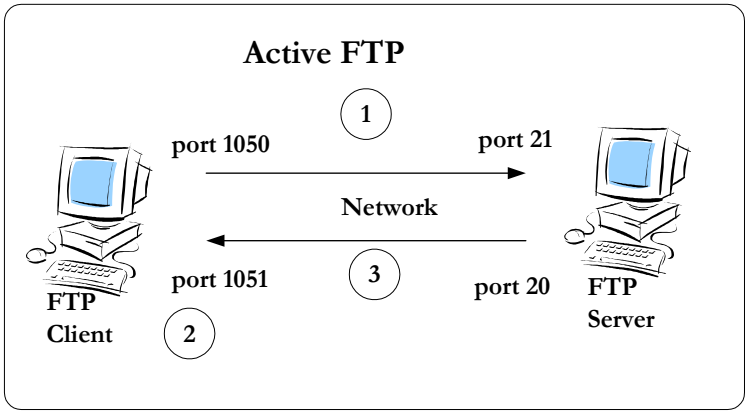
Packet จาก INPUT Chain ที่เกิดจากการติดต่อที่เกิดขึ้นอยู่แล้วหรือการติดต่อที่สัมพันธ์กันจะถูกส่งผ่านไปได้

การ Filter Packe ของ FTP Service

FTP (File Transfer Protocol) ทำงานที่ Port 21 เป็น Port ที่ส่งคำสั่งควบคุม (Control Port) ส่วน Data ที่ Upload หรือ Download ส่งผ่านอีก Port หนึ่งเรียกว่า ftp-data (Port 20) แบ่งการทำงานเป็น 2 แบบคือ Active และ Passive

```
iptables -A INPUT     -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Active FTP



command client(>1024) -> server (21)

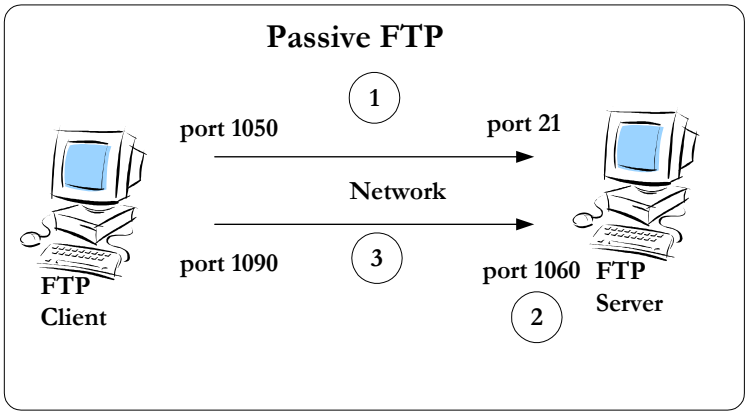
data client(>1024) <- server (20)

FTP Client จะส่งหมายเลข Port ที่ต้องการรับ Data ไปให้ FTP Server (ผ่าน Port 21 ของ Server) เมื่อ Server ได้รับ Port สำหรับรับ Data ของ Client แล้ว Server จะส่งข้อมูลจาก Port 20 ของ Server ไปยัง Client โดย Server จะขอเปิดการติดต่อใหม่ไปยัง Client

กำหนด rule ให้กับ INPUT และ OUTPUT Chain ได้ดังนี้

```
iptables -A INPUT     -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
```


Passive FTP



command client (>1024) -> server (21)

data client (>1024) -> server (>1024)

FTP Client ติดต่อไปยัง FTP Server ผ่าน Port 21 แล้วส่งคำสั่งให้ Server เปิด Port ขึ้นมาใหม่อีก Port หนึ่ง เป็น Nonpriviledge Port (ค่ามากกว่า 1023) จากนั้น Server จะส่งหมายเลข Port ไปบอก Client เพื่อให้ Client ติดต่อรับส่ง Data

```
iptables -A INPUT     -p tcp --sport 1024: --dport 1024:  -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1024: --dport 1024:  -m state --state ESTABLISHED,RELATED -j ACCEPT
```

NAT (Network Address Translation)

DNAT (Destination Network Address Translation)

เป็นการแปลง Destination IP Address จาก IP Address ของ Firewall ให้เป็น IP Address ของเครื่องที่อยู่ Network ภายใน ใช้ในกรณีต้องการ Redirect Packet ที่ส่งมายัง Firewall ไปยังเครื่องใน Network ภายใน เครื่องจากภายนอกจะติดต่อกับ Firewall เท่านั้น

SNAT (Source Network Address Translation)

เป็นการแปลง Source IP Address ไปเป็น IP Address ของ Firewall (เป็น IP Address ที่แน่นอนและต้องรู้ค่า IP Address ก่อน) ก่อนส่งออกไปยัง Internet

MASQUERADE

หลักการทำงานเหมือนกันกับ SNAT แต่เหมาะกับการทำงานกับ Interface ที่เป็น Dialup Connection (PPP, SLIP) หรือ DHCP เพราะ Masquerade สนใจ Interface (ไม่จำเป็นต้องรู้ค่า IP Address ก่อน) เวลาทำงานจึงใช้ IP Address ของ Interface นั้นๆ

table nat ประกอบด้วย 3 Chain

PREROUTING	Packet ที่รับเข้ามา
POSTROUTE	Packet ที่ส่งออกไป (เป็น Packet ที่รับเข้าซึ่งผ่านขั้นตอนการ Routing แล้ว)
OUTPUT	Packet ที่เริ่มต้นออกจากเครื่อง

เวลาเรียกใช้งานต้องกำหนดชื่อของ table โดยใช้ Option -t nat

```
# iptables -t nat -nL
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

SNAT กำหนดที่ Chain POSTROUTING

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 194.236.50.155-
194.236.50.160:1024-32000
```

MASQUERADE กำหนดที่ Chain POSTROUTING

```
iptables -t nat -A POSTROUTING -p TCP  -j MASQUERADE --to-ports 1024-31000
```

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

แสดงสถานะของ Address Translation

```
# iptables -t nat -L -v
Chain PREROUTING (policy ACCEPT 3727 packets, 478K bytes)
 pkts bytes target     prot opt in     out     source
destination

Chain POSTROUTING (policy ACCEPT 494 packets, 49025 bytes)
 pkts bytes target     prot opt in     out     source
destination
   84  5677 MASQUERADE  all  --  any    eth1    anywhere
anywhere

Chain OUTPUT (policy ACCEPT 416 packets, 44466 bytes)
 pkts bytes target     prot opt in     out     source
destination
```

DNAT กำหนดที่ Chain PREROUTING

```
iptables -t nat -A PREROUTING -p tcp -d 160.10.1.1 --dport 80 -j DNAT --to-destination
192.168.1.1-192.168.1.10
```

REDIRECT

กำหนดที่ Chain PREROUTING

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 3128
```

ตัวอย่างการทำงานของ Transparence Proxy

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 3128
iptables -t nat -A POSTROUTING  -j MASQUERADE
```

LOG

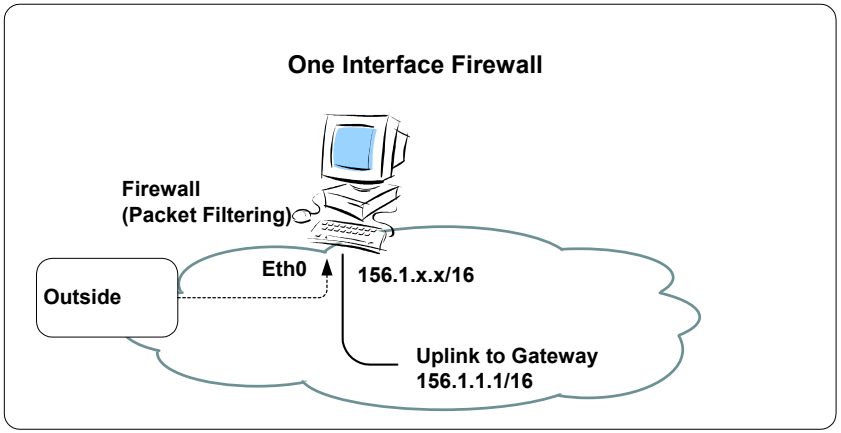
```
iptables -A FORWARD -p tcp -j LOG --log-level debug --log-ip-options --log-tcp-options
--log-tcp-sequence --log-prefix "input packet"
```


Shorewall

เป็น Software ที่ทำงานอยู่บน iptables ทำให้สามารถกำหนดค่าการทำงานของ Firewall ได้ง่ายขึ้น ดูรายละเอียดได้จาก http://www.shorewall.net/

One Interface Firewall

ใช้ในกรณีที่เครื่อง Server ต่อเข้ากับ Internet แล้วต้องการควบคุมการติดต่อ



Shorewall เก็บค่าเริ่มต้นการทำงานไว้ใน Directory /etc/shorewall โดยแบ่งเป็น file ซึ่งมีรายละเอียดดังนี้

กำหนด Zone

/etc/shorewall/zones

#ZONE	DISPLAY	COMMENTS
net	Net	Internet

เป็นการกำหนด zone ชื่อว่า net ใช้แทนเครื่องต่างๆ จาก internet ซึ่งจะถูกนำไปกำหนด policy ต่อไป

สำหรับเครื่องที่เป็น firewall เองจะมี Zone ชื่อว่า fw โดยไม่ต้องกำหนดใน File zone

กำหนด Interface

/etc/shorewall/interfaces

```
#ZONE      INTERFACE  BROADCASTOPTIONS
net        eth0       detect
```

Interface eth0 ต่ออยู่กับ Zone net

กำหนด Policy

/etc/shorewall/policy

```
#SOURCE        DEST        POLICY        LOG LEVEL
LIMIT:BURST
fw             net         ACCEPT
net            all         DROP          info
all            all         REJECT        info
```

การติดต่อจากเครื่อง firewall ไปยัง internet ทำได้

การติดต่อจาก Internet มายัง firewall ถูก Drop

การติดต่อที่เหลือ all จะถูก Reject

กำหนดเงื่อนไขการติดต่อ

/etc/shorewall/rules

```
#ACTION   SOURCE   DESTINATION   PROTOCOL   PORT   SOURCE PORT   ORIGINAL ADDRESS
ACCEPT    net      fw            tcp        22
ACCEPT    net      fw            tcp        80
ACCEPT    net      fw            tcp        110
ACCEPT    net      fw            tcp        25
```

รายละเอียดของ Service เก็บไว้ใน file /etc/services

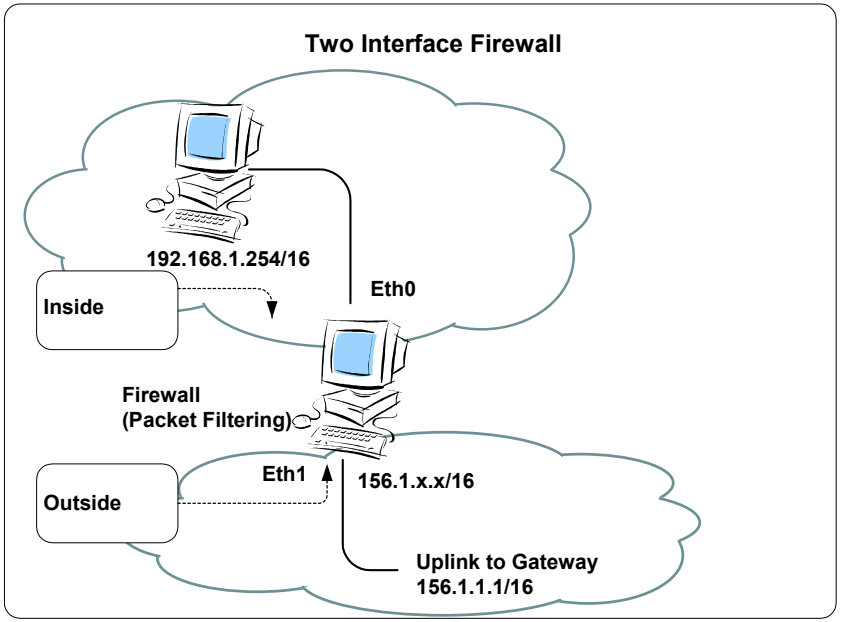
เมื่อกำหนดค่าเริ่มต้นใน File ต่างๆ ของ Shorewall เรียบร้อยแล้วต้องสั่งให้ Service ของ Shorewall Restart ด้วยคำสั่ง

```
# service shorewall restart
```

ตรวจสอบการทำงานด้วยคำสั่ง

```
# iptables -L
```

Two Interface Firewall



กำหนด Zone

/etc/shorewall/zones

#ZONE	DISPLAY	COMMENTS
net	Net	Internet
loc	Local	Local networks

net แทนเครื่องจาก internet

loc แทนเครื่องจาก local network

กำหนด Interface

/etc/shorewall/interfaces

#ZONE	INTERFACE	BROADCASTOPTIONS
net	eth0	detect
loc	eth1	detect

กำหนด Policy

/etc/shorewall/policy

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT:BURST
loc	net	ACCEPT		
fw	net	ACCEPT		
net	all	DROP	info	
all	all	REJECT	info	

กำหนดเงื่อนไขการติดต่อ

/etc/shorewall/rules

#ACTION	SOURCE	DESTINATION	PROTOCOL	PORT	SOURCE PORT	ORIGINAL ADDRESS
ACCEPT	loc	fw	tcp	53		
ACCEPT	loc	fw	udp	53		
ACCEPT	loc	fw	tcp	22		
ACCEPT	loc	fw	tcp	80		
ACCEPT	net	fw	tcp	80		
ACCEPT	net	fw	tcp	25		

NAT และ Masquerade

/etc/shorewall/masq

```
#INTERFACE      SUBNET          ADDRESS
eth0            eth1
```

Packet ที่มาจาก Interface eth1 จะถูก masquerade (เปลี่ยน Source Address) ด้วย IP Address ของ Interface eth0

ส่วน Address ใช้กำหนด IP Address สำหรับทำ SNAT

/etc/shorewall/shorewall.conf

เปลี่ยนค่าของ Option

```
NAT_ENABLED=Yes
IP_FORWARDING=On
```

กำหนดเงื่อนไขการติดต่อ

/etc/shorewall/rules

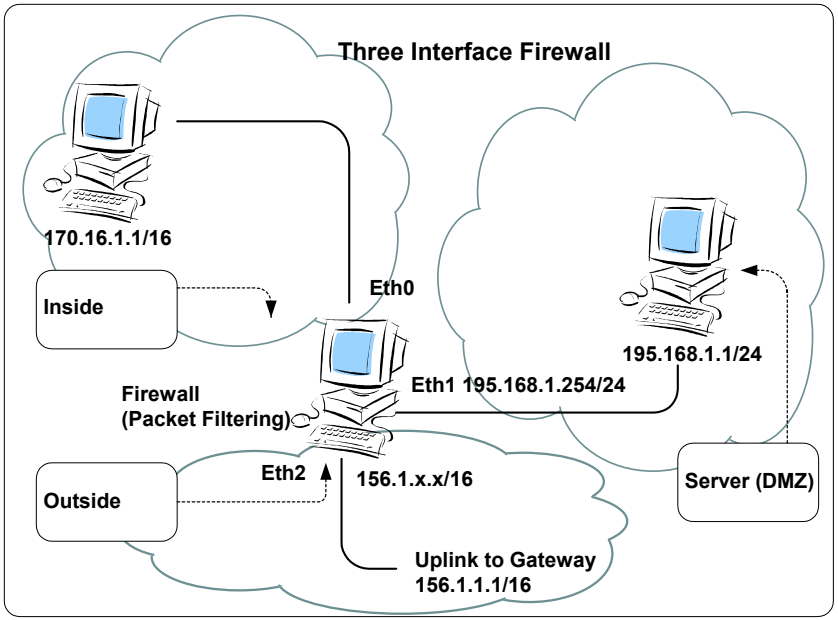
#ACTION	SOURCE	DESTINATION	PROTOCOL	PORT	SOURCE PORT	ORIGINAL ADDRESS
ACCEPT	loc	fw	tcp	22		
DNAT	net	loc:192.168.1.1	tcp	80		
REDIRECT	loc	3128	tcp	80	-	!192.168.1.254

รับการติดต่อจาก Client จาก local network มายัง Service ของ Secure Shell

รับการติดต่อจาก Client จาก Internet ที่ส่งมายัง Port 80 ของ Firewall แล้ว Redirect Packet นั้นไปยังเครื่อง ใน local network ที่มี IP Address 192.168.1.1

รับการติดต่อจาก Client จาก local network ที่ส่งไปยัง Port 80 แล้ว Redirect ไปยัง Port 3128 ของเครื่อง Firewall (ใช้ทำ Transparence Proxy)

Three Interface Firewall



/etc/shorewall/zones

#ZONE	DISPLAY	COMMENTS
net	Net	Internet
loc	Local	Local networks
dmz	DMZ	Demilitarized Zone

/etc/shorewall/interfaces

```
#ZONE      INTERFACE  BROADCAST OPTIONS
net    eth0        detect
loc    eth1        detect      routestopped
dmz    eth2        detect      routestopped
```

/etc/shorewall/routestopped

```
#INTERFACE HOST(S)
eth1       -
eth2       -
```

/etc/shorewall/policy

```
#SOURCE     DEST     POLICY     LOG LEVEL   LIMIT:BURST
loc         net      ACCEPT
net         all      DROP       info
all         all      REJECT     info
```

/etc/shorewall/masq

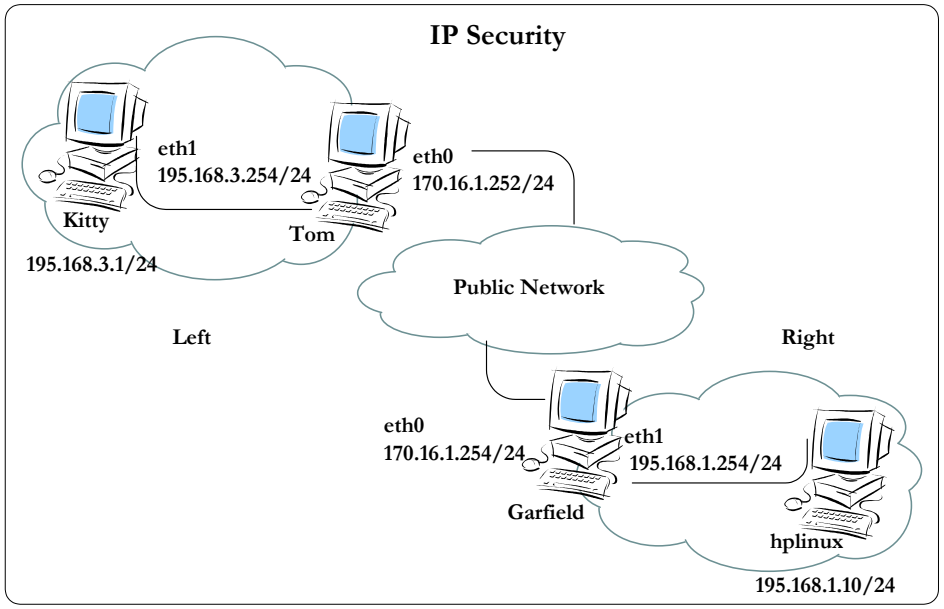
```
#INTERFACE     SUBNET          ADDRESS
eth0           eth1
eth0           eth2
```


/etc/shorewall/rules

```
#ACTION   SOURCE   DESTINATION PROTOCOL PORT  SOURCE  ORIGINAL
                                              PORT    ADDRESS
ACCEPT    fw       net         tcp      53
ACCEPT    fw       net         udp      53

ACCEPT    loc      fw          tcp      22
ACCEPT    loc      dmz         tcp      22

ACCEPT    dmz      net         tcp      53
ACCEPT    dmz      net         udp      53
```

IPSEC Tunnel

สำหรับทำงานร่วมกับ Freeswan



Network A

/etc/shorewall/tunnels

# TYPE	ZONE	GATEWAY	GATEWAY ZONE
ipsec	net	195.168.1.254	

Network B

/etc/shorewall/tunnels

# TYPE	ZONE	GATEWAY	GATEWAY ZONE
ipsec	net	170.16.1.254	

เพิ่ม zone vpn

/etc/shorewall/zones

#ZONE	DISPLAY	COMMENTS
vpn	VPN	Remote Subnet

เพิ่ม interface ipsec0

/etc/shorewall/interfaces

#ZONE	INTERFACE	BROADCAST	OPTIONS
vpn	ipsec0		

เพิ่ม policy

/etc/shorewall/policy

#SOURCE	DEST	POLICY	LOG LEVEL	LIMIT:BURST
loc	vpn	ACCEPT		
vpn	loc	ACCEPT		

ขั้นตอนการ Compile Kernel บน FreeBSD เพื่อให้สนับสนุน ISL

สร้าง Directory สำหรับเก็บ Source Code

```
#mkdir /usr/local/src
#cd /usr/local/src
```

Download File จาก ftp://ftp.radio-msu.net/pub/homebrew/FreeBSD/isl-0.2.1.tgz

```
#gzip -cd isl-0.2.4.tgz |tar xvf -
#cd isl-0.2.4/FreeBSD-4.4
```

Copy file if_isl.c และ if_isl_var.h ไว้ใน Directory /usr/src/sys/net

```
#cp if_isl.c  /usr/src/sys/net/
#cp if_isl_var.h /usr/src/sys/net/
```

Patch File

/usr/src/sys/conf/files

```
#patch < files.diff
Hmm...  Looks like a new-style context diff to me...
The text leading up to this was:
--------------------------
|*** /usr/src/sys/conf/files.orig       Tue Nov 20 10:55:38 2001
|--- /usr/src/sys/conf/files    Tue Nov 20 10:56:13 2001
--------------------------
File to patch: /usr/src/sys/conf/files
Patching file /usr/src/sys/conf/files using Plan A...
Hunk #1 succeeded at 638 (offset -40 lines).
done
```


/usr/src/sys/conf/options

```
#patch < options.diff
Hmm...  Looks like a new-style context diff to me...
The text leading up to this was:
--------------------------
|*** /usr/src/sys/conf/options.orig     Tue Nov 20 10:56:30 2001
|--- /usr/src/sys/conf/options  Tue Nov 20 10:57:14 2001
--------------------------
File to patch: /usr/src/sys/conf/options
Patching file /usr/src/sys/conf/options using Plan A...
Hunk #1 succeeded at 445 (offset -16 lines).
done
```

/usr/src/sys/net/if_ethersubr.c

```
#patch <if_ethersubr.c.diff
Hmm...  Looks like a new-style context diff to me...
The text leading up to this was:
--------------------------
|*** /usr/src/sys/net/if_ethersubr.c.orig       Tue Nov 20 10:57:42
2001
|--- /usr/src/sys/net/if_ethersubr.c    Tue Nov 20 11:01:41 2001
--------------------------
File to patch: /usr/src/sys/net/if_ethersubr.c
Patching file /usr/src/sys/net/if_ethersubr.c using Plan A...
Hunk #1 succeeded at 104.
Hunk #2 succeeded at 526.
done
```

Config Kernel

Directory ที่เก็บ Source Code ของ Kernel อยู่ใน /usr/src/sys

กำหนด Option สำหรับ Kernel ที่จะ Compile เข้าไปใน Directory /usr/src/sys/i386/conf

Copy Kernel Configuration จาก GENERIC มาเป็น File ที่ต้องการแก้ไข (ISLKERNEL)

```
#cd /usr/src/sys/i386/conf
#cp GENERIC ISLKERNEL
```

แก้ไข File ISLKERNEL เพิ่ม Option ต่อท้าย File

```
pseudo-device    isl   4
```

4 คือจำนวน Interface (VLAN) ที่ต้องการ

ใช้คำสั่ง config <Kernel Config File> เพื่อสร้าง File ที่จำเป็นสำหรับการ Compile Kernel

```
#config ISLKERNEL
Don't forget to do a ``make depend''
Kernel build directory is ../../compile/ISLKERNEL
```

เข้าไปใน Directory ที่เก็บ Source File ของ Kernel

```
#cd ../../compile/ISLKERNEL
#make depend
#make
#make install
```


Reboot Server

```
#reboot
```

หลังจาก Reboot แล้ว Server จะใช้ Kernel ตัวใหม่ที่สนับสนุน ISL Encapsulation

คำสั่ง ifconfig ที่ใช้กำหนดค่าสำหรับ Interface จำเป็นต้อง Compile ใหม่เพื่อให้สนับสนุนกับ Interface ที่เป็น ISL

```
#cd /usr/local/src/isl-2.0.4
#cd ifconfig-FreeBSD-4.2
#make
```

เมื่อ make เรียบร้อยแล้วจะได้ file ifconfig ให้ copy ไปไว้ที่ Directory /sbin

```
#mv /sbin/ifconfig /sbin/ifconfig.org
#cp ifconfig /sbin/ifconfig
#chmod 755 /sbin/ifconfig
```

เรียกใช้คำสั่ง ifconfig ตรวจสอบสถานะของ Network Interface

```
# ifconfig
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::202:44ff:fe0c:c11b%rl0 prefixlen 64 scopeid 0x1
        ether 00:02:44:0c:c1:1b
        media: 100baseTX <full-duplex> status: active
        supported media: autoselect 100baseTX <full-duplex> 100baseTX
10baseT/UTP <full-duplex> 10baseT/UTP 100baseTX <hw-loopback>
isl0: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0, parent: <none>, encapsulated frame: Ethernet, priority: Normal
isl1: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0, parent: <none>, encapsulated frame: Ethernet, priority: Normal
isl2: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0, parent: <none>, encapsulated frame: Ethernet, priority: Normal
isl3: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0, parent: <none>, encapsulated frame: Ethernet, priority: Normal
```

ifconfig <ifname> media 100baseTX mediaopt full-duplex up

ifconfig <isl no> inet <ip address> netmask <netmask> isldev <ifname> islvlan <vlan no>

<ifname> Interface Name แสดงจากคำสั่ง ifconfig ตัวอย่างเช่น rl0

<isl no> ชื่อของ Interface isl0 isl1 isl2 isl3 (มี 4 Interface ตาม Option ที่กำหนดก่อน Compile Kernel แสดงผลได้จากคำสั่ง ifconfig)

<vlan no> หมายเลข vlan 1-1000

ใช้งาน Interface ที่กับ Trunk Port ของ Switch (ไม่ต้องกำหนด IP Address)

#ifconfig rl0 media 100baseTX mediaopt full-duplex up

กำหนด IP Address ให้กับ Sub Interface ด้วยคำสั่ง

```
#ifconfig isl0 inet 192.168.1.254 netmask 255.255.255.0 isldev rl0 islvlan 1
#ifconfig isl1 inet 192.168.2.254 netmask 255.255.255.0 isldev rl0 islvlan 2
#ifconfig isl2 inet 192.168.3.254 netmask 255.255.255.0 isldev rl0 islvlan 3
#ifconfig isl3 inet 192.168.4.254 netmask 255.255.255.0 isldev rl0 islvlan 4
```

ตรวจสอบด้วยคำสั่ง ifconfig

```
#ifconfig
isl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.1.254 netmask 0xffffff00 broadcast 192.168.1.255
        ether 00:02:44:0c:c1:1c
        vlan: 1, parent: rl0, encapsulated frame: Ethernet, priority: Normal
isl1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.2.254 netmask 0xffffff00 broadcast 192.168.2.255
        ether 00:02:44:0c:c1:1d
        vlan: 2, parent: rl0, encapsulated frame: Ethernet, priority: Normal
isl2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.3.254 netmask 0xffffff00 broadcast 192.168.3.255
        ether 00:02:44:0c:c1:1e
        vlan: 3, parent: rl0, encapsulated frame: Ethernet, priority: Normal
isl3: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.4.254 netmask 0xffffff00 broadcast 192.168.4.255
        ether 00:02:44:0c:c1:1f
        vlan: 4, parent: rl0, encapsulated frame: Ethernet, priority: Normal
```

ping ไปที่ IP Address ของ Sub Interface และแสดง Routing Table ด้วยคำสั่ง netstat -rn

```
# netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags     Refs     Use     Netif Expire
127.0.0.1          127.0.0.1          UH          0       0      lo0
192.168.1          link#11            UC          0       0     isl0 =>
192.168.2          link#12            UC          0       0     isl1 =>
192.168.3          link#13            UC          0       0     isl2 =>
192.168.4          link#14            UC          0       0     isl3 =>

Internet6:
Destination                       Gateway                       Flags
Netif Expire
::1                               ::1                           UH          lo0
fe80::%rl0/64                     link#1                        UC          rl0
fe80::%lo0/64                     fe80::1%lo0                   Uc          lo0
ff01::/32                         ::1                           U           lo0
ff02::%rl0/32                     link#1                        UC          rl0
ff02::%lo0/32                     fe80::1%lo0                   UC          lo0
```

กำหนดค่าเริ่มต้นของ Interface ใน File /etc/rc.conf เพื่อใช้งานทุกครั้งเมื่อ Boot

```
# -- sysinstall generated deltas -- #
# Created: Wed Feb 26 16:05:59 2003
# Enable network daemons for user convenience.
# This file now contains just the overrides from /etc/defaults/rc.conf
# please make all changes to this file.
inetd_enable="YES"
kern_securelevel_enable="NO"
sendmail_enable="YES"
sshd_enable="YES"

ifconfig_rl0="media 100baseTX mediaopt full-duplex up"
ifconfig_isl0="inet 192.168.1.254 netmask 255.255.255.0 isldev rl0 islvlan 1"
ifconfig_isl1="inet 192.168.2.254 netmask 255.255.255.0 isldev rl0 islvlan 2"
ifconfig_isl2="inet 192.168.3.254 netmask 255.255.255.0 isldev rl0 islvlan 3"
ifconfig_isl3="inet 192.168.4.254 netmask 255.255.255.0 isldev rl0 islvlan 4"
```

Reboot เครื่อง Server และหลังจาก Reboot เรียบร้อยแล้วให้ตรวจสอบสถานะของ Interface ด้วยคำสั่ง ifconfig

Config Catalyst 1900

เข้า Config Switch ผ่าน Console หรือ Telnet

```
Catalyst 1900 Management Console
Copyright (c) Cisco Systems, Inc.  1993-1999
All rights reserved.
Enterprise Edition Software
Ethernet Address:      00-08-A3-BE-2B-00

PCA Number:            73-3124-04
PCA Serial Number:     FAB060543YZ
Model Number:          WS-C1924C-EN
System Serial Number:  FAB0605Y0NE
Power Supply S/N:      APR050603DC
PCB Serial Number:     FAB060543YZ,73-3124-04
-------------------------------------------------

1 user(s) now active on Management Console.

        User Interface Menu

     [M] Menus
     [K] Command Line

Enter Selection:  K


      CLI session with the switch is open.
      To end the CLI session, enter [Exit].
```

ที่ Config Mode สร้าง VLAN 2 ถึง 4 (VLAN 1 มีอยู่แล้วเป็น Default ไม่ต้องสร้าง) รูปแบบของคำสั่ง vlan <vlan no> <type>

```
Cat1924F>enable
Enter password:  ******
Cat1924F#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z
Cat1924F(config)#vlan 2 ethernet
Cat1924F(config)#vlan 3 ethernet
Cat1924F(config)#vlan 4 ethernet
```

กำหนดให้แต่ละ Port ของ Switch เป็นสมาชิกของ VLAN ที่สร้างขึ้น โดยกำหนดให้หมายเลข Port ตรงกับหมายเลขของ VLAN

```
Cat1924F(config)#interface ethernet 0/1
Cat1924F(config-if)#vlan-membership static 1
Cat1924F(config-if)#exit
Cat1924F(config)#interface ethernet 0/2
Cat1924F(config-if)#vlan-membership static 2
Cat1924F(config-if)#exit
Car1924F(config)#interface ethernet 0/3
Cat1924F(config-if)#vlan-membership static 3
Cat1924F(config-if)#exit
Cat1924F(config)#interface ethernet 0/4
Cat1924F(config-if)#vlan-membership static 4
^Z
```

แสดง VLAN และ Port ที่เป็นสมาชิกของ VLAN ด้วยคำสั่ง show vlan

```
CAT1924F#show vlan
VLAN Name              Status     Ports
-------------------------------------
1    default           Enabled    1, AUI, A, B
2    VLAN0002          Enabled    2
3    VLAN0003          Enabled    3
4    VLAN0004          Enabled    4
1002 fddi-default      Suspended
1003 token-ring-defau  Suspended
1004 fddinet-default   Suspended
1005 trnet-default     Suspended
-------------------------------------
```

กำหนดให้ Port B เป็น Trunk Port ด้วยคำสั่ง trunk on

```
Cat1924F(config)#int fastEthernet 0/27
Cat1924F(config-if)#trunk on
^Z
Cat1924F#show trunk b
DISL state: On, Trunking: On, Encapsulation type: ISL
```


One Arm Router โดยใช้ FreeBSD และ 802.1Q

Config Kernel

Directory ที่เก็บ Source Code ของ Kernel อยู่ใน /usr/src/sys

กำหนด Option สำหรับ Kernel ที่จะ Compile เข้าไปใน Directory /usr/src/sys/i386/conf

Copy Kernel Configuration จาก GENERIC มาเป็น File ที่ต้องการแก้ไข (DOT1QKERNEL)

```
#cd /usr/src/sys/i386/conf
#cp GENERIC DOT1QKERNEL
```

แก้ไข File DOT1QKERNEL เพิ่ม Option ต่อท้าย File

```
pseudo-device    vlan  4
```

4 คือจำนวน Interface (VLAN) ที่ต้องการ

ใช้คำสั่ง config <Kernel Config File> เพื่อสร้าง File ที่จำเป็นสำหรับการ Compile Kernel

```
#config DOT1QKERNEL
Don't forget to do a ``make depend''
Kernel build directory is ../../compile/DOT1QKERNEL
```

เข้าไปใน Directory ที่เก็บ Source File ของ Kernel

```
#cd ../../compile/DOT1QKERNEL
#make depend
#make
#make install
```

Reboot Server

```
#reboot
```

หลังจาก Reboot แล้ว Server จะใช้ Kernel ตัวใหม่ที่สนับสนุน 802.1Q Encapsulation

เรียกใช้คำสั่ง ifconfig ตรวจสอบสถานะของ Network Interface

```
# ifconfig
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        inet6 fe80::202:44ff:fe0c:c11b%rl0 prefixlen 64 scopeid 0x1
        ether 00:02:44:0c:c1:1b
        media: 100baseTX <full-duplex> status: active
        supported media: autoselect 100baseTX <full-duplex> 100baseTX
10baseT/UTP <full-duplex> 10baseT/UTP 100baseTX <hw-loopback>
vlan0: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0 parent interface: <none>
vlan1: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0 parent interface: <none>
vlan2: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0 parent interface: <none>
vlan3: flags=0<> mtu 1500
        ether 00:00:00:00:00:00
        vlan: 0 parent interface: <none>
```

ifconfig <ifname> media 100baseTX mediaopt full-duplex up

ifconfig <vlan no> inet <ip address> netmask <netmask> vlan <no> vlandev <ifname>

<ifname> Interface Name แสดงจากคำสั่ง ifconfig ตัวอย่างเช่น rl0

<vlan no> ชื่อของ Interface vlan0 vlan1 vlan2 vlan3 (มี 4 Interface ตาม Option ที่กำหนด ก่อน Compile Kernel แสดงผลได้จากคำสั่ง ifconfig)

<no> หมายเลข vlan 1-1000

ใช้งาน Interface ที่กับ Trunk Port ของ Switch (ไม่ต้องกำหนด IP Address)

#ifconfig rl0 media 100baseTX mediaopt full-duplex up

กำหนด IP Address ให้กับ Sub Interface ด้วยคำสั่ง

```
#ifconfig vlan0 inet 192.168.1.254 netmask 255.255.255.0 vlan 1 vlandev rl0
#ifconfig vlan1 inet 192.168.2.254 netmask 255.255.255.0 vlan 2 vlandev rl0
#ifconfig vlan2 inet 192.168.3.254 netmask 255.255.255.0 vlan 3 vlandev rl0
#ifconfig vlan3 inet 192.168.4.254 netmask 255.255.255.0 vlan 4 vlandev rl0
```

ตรวจสอบด้วยคำสั่ง ifconfig

```
#ifconfig
vlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1496
        inet 192.168.1.254 netmask 0xffffff00 broadcast 192.168.1.255
        inet6 fe80::202:44ff:fe0c:c11b%vlan0 prefixlen 64 scopeid 0x5
        ether 00:02:44:0c:c1:1b
        vlan: 1 parent interface: rl0
vlan1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1496
        inet 192.168.2.254 netmask 0xffffff00 broadcast 192.168.2.255
        inet6 fe80::202:44ff:fe0c:c11b%vlan1 prefixlen 64 scopeid 0x6
        ether 00:02:44:0c:c1:1b
        vlan: 2 parent interface: rl0
vlan2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1496
        inet 192.168.3.254 netmask 0xffffff00 broadcast 192.168.3.255
        inet6 fe80::202:44ff:fe0c:c11b%vlan2 prefixlen 64 scopeid 0x7
        ether 00:02:44:0c:c1:1b
        vlan: 3 parent interface: rl0
vlan3: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1496
        inet 192.168.4.254 netmask 0xffffff00 broadcast 192.168.4.255
        inet6 fe80::202:44ff:fe0c:c11b%vlan3 prefixlen 64 scopeid 0x8
        ether 00:02:44:0c:c1:1b
        vlan: 4 parent interface: rl0
```

ping ไปที่ IP Address ของ Sub Interface และแสดง Routing Table ด้วยคำสั่ง netstat -rn

```
# netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags      Refs      Use      Netif Expire
127.0.0.1          127.0.0.1          UH          0        0       lo0
192.168.1          link#5             UC          0        0    vlan0 =>
192.168.2          link#6             UC          0        0    vlan1 =>
192.168.3          link#7             UC          0        0    vlan2 =>
192.168.4          link#8             UC          0        0    vlan3 =>
```

กำหนดค่าเริ่มต้นของ Interface ใน File /etc/rc.conf เพื่อใช้งานทุกครั้งเมื่อ Boot

```
# -- sysinstall generated deltas -- #
# Created: Wed Feb 26 16:05:59 2003
# Enable network daemons for user convenience.
# This file now contains just the overrides from /etc/defaults/rc.conf
# please make all changes to this file.
inetd_enable="YES"
kern_securelevel_enable="NO"
sendmail_enable="YES"
sshd_enable="YES"

ifconfig_rl0="media 100baseTX mediaopt full-duplex up"
ifconfig_vlan0="inet 192.168.1.254 netmask 255.255.255.0 vlan 1 vlandev rl0"
ifconfig_vlan1="inet 192.168.2.254 netmask 255.255.255.0 vlan 2 vlandev rl0"
ifconfig_vlan2="inet 192.168.3.254 netmask 255.255.255.0 vlan 3 vlandev rl0"
ifconfig_vlan3="inet 192.168.4.254 netmask 255.255.255.0 vlan 4 vlandev rl0"
```

Reboot เครื่อง Server และหลังจาก Reboot เรียบร้อยแล้วให้ตรวจสอบสถานะของ Interface ด้วยคำสั่ง ifconfig

Config Catalyst 2924XL

เข้า Config Switch ผ่าน Console หรือ Telnet

สร้าง VLAN โดยใช้คำสั่ง vlan database แล้วเรียกใช้คำสั่ง vlan <vlan number>

```
User Access Verification

Password:
c2924>enable
c2924#vlan database
C2924(vlan)#vlan 2
VLAN 2 added:
    Name: VLAN0002
C2924(vlan)#vlan 3
VLAN 3 added:
    Name: VLAN0003
C2924(vlan)#vlan 4
VLAN 4 added:
    Name: VLAN0004
C2924(vlan)#exit
APPLY completed.
Exiting....
```

แสดงรายละเอียดของ VLAN ที่สร้าง ด้วยคำสั่ง show vlan

```
C2924#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
2    VLAN0002                         active
3    VLAN0003                         active
4    VLAN0004                         active
```

กำหนดให้ Port ของ Switch เป็นสมาชิกของ VLAN ที่สร้างไว้ โดยเข้าไปที่ Port ของ Switch ด้วยคำสั่ง interface <interfacename> แล้วใช้คำสั่ง switchport access vlan <vlan number>

```
C2924# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
C2924(config)#interface fastEthernet 0/1
C2924(config-if)# switchport access vlan 1
C2924(config-if)#exit
C2924(config)#interface fastEthernet 0/2
C2924(config-if)# switchport access vlan 2
C2924(config-if)#exit
C2924(config)#interface fastEthernet 0/3
C2924(config-if)# switchport access vlan 3
C2924(config-if)#exit
C2924(config)#interface fastEthernet 0/4
C2924(config-if)# switchport access vlan 4
C2924(config-if)#exit
```

กำหนดให้ Port ของ Switch (port 24) ทำงานใน mode trunk ด้วยคำสั่ง switchport mode trunk และเลือก trunk encapsulation เป็น dot1q (IEEE802.1Q) ด้วยคำสั่ง switchport trunk encapsulation dot1q

```
C2924(config)#interface fastEthernet0/24
C2924(config-if)# switchport trunk encapsulation dot1q
C2924(config-if)# switchport mode trunk
C2924(config-if)#
^z
```

แสดงรายละเอียดของ VLAN หลังจากกำหนดให้ Port ต่างๆ เป็นสมาชิกแล้ว ด้วยคำสั่ง show vlan

```
C2924#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/5, Fa0/6, Fa0/7,
                                                Fa0/8, Fa0/9, Fa0/10, Fa0/11,
                                                Fa0/12,Fa0/13, Fa0/14, Fa0/15,
                                                Fa0/16,Fa0/17, Fa0/18, Fa0/19,
                                                Fa0/20,Fa0/21, Fa0/22, Fa0/23,
                                                Fa0/24
2    VLAN0002                         active    Fa0/2
3    VLAN0003                         active    Fa0/3
4    VLAN0004                         active    Fa0/4
```


แสดงรายละเอียดของ Port ต่างๆ ด้วยคำสั่ง show interfaces status

```
C2924#show interfaces status

Port     Name               Status       Vlan     Duplex Speed   Type
-------- ------------------ ------------ -------- ------ ------- ----
Fa0/1                       connected    1        A-Half    A-10 100BaseTX/FX
Fa0/2                       connected    2        A-Half   A-100 100BaseTX/FX
Fa0/3                       notconnect   3          Auto    Auto 100BaseTX/FX
Fa0/4                       notconnect   4          Auto    Auto 100BaseTX/FX
.
.
Fa0/24                      connected  Trunk      A-Full   A-100 100BaseTX/FX
```

One Arm Router โดยใช้ Linux และ 802.1Q

http://www.candelatech.com/~greear/vlan/cisco_howto.html

Kernel 2.4 ของ Linux จะสนับสนุนมาตรฐาน 802.1Q อยู่แล้วแต่ Kernel Module ยังไม่มีการเรียกใช้งาน ตรวจสอบ Module ด้วยคำสั่ง lsmod

```
# lsmod
Module                  Size  Used by    Not tainted
ppp_async               7456   0  (unused)
ppp_generic            20064   0  [ppp_async]
slhc                    5072   0  [ppp_generic]
ip_vs                  74328   0  (autoclean)
af_packet              13000   0  (autoclean)
ne                      6544   1  (autoclean)
8390                    6192   0  (autoclean) [ne]
8139too                14472   1  (autoclean)
# modprobe 8021q
# lsmod
Module                  Size  Used by    Not tainted
8021q                  13832   0  (unused)
ppp_async               7456   0  (unused)
ppp_generic            20064   0  [ppp_async]
slhc                    5072   0  [ppp_generic]
ip_vs                  74328   0  (autoclean)
af_packet              13000   0  (autoclean)
ne                      6544   1  (autoclean)
8390                    6192   0  (autoclean) [ne]
8139too                14472   1  (autoclean)
```

เพิ่มชื่อ module ที่ต้องการเรียกขึ้นมาใช้งานเวลาเครื่อง Server Boot ไว้ใน File /etc/modules

```
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line.  Comments begin with
# a `#', and everything on the line after them are ignored.
8021q
```


Download Software จัดการ VLAN จาก

http://www.candelatech.com/~greear/vlan/vlan.1.6.tar.gz เก็บไว้ใน Directory /usr/local/src/

```
# pwd
/usr/local/src
# ls
vlan.1.6.tar.gz
#
# gzip -cd vlan.1.6.tar.gz |tar xvf -
# ls
vlan/  vlan.1.6.tar.gz
# cd vlan
# make
# cp vconfig /sbin/
```

Startup Interface eth0 ด้วยคำสั่ง ifconfig (ไม่ต้องกำหนด IP Address)

```
# ifconfig eth0 inet 0.0.0.0 up
```

เพิ่ม VLAN ด้วยคำสั่ง vconfig add eth0 <vlan no>

```
# vconfig add eth0 1
Added VLAN with VID == 1 to IF -:eth0:-
WARNING:  VLAN 1 does not work with many switches,
consider another number if you have problems.
# vconfig add eth0 2
Added VLAN with VID == 2 to IF -:eth0:-
# vconfig add eth0 3
Added VLAN with VID == 3 to IF -:eth0:-
# vconfig add eth0 4
Added VLAN with VID == 4 to IF -:eth0:-
```

ตรวจสอบ Interface ที่เพิ่มด้วยคำสั่ง ifconfig -a (-a จะแสดง Interface ทั้งที่ Start และ Shutdown) ได้ Interface eth0.1 eth0.2 eth0.3 eth0.4 เพิ่มขึ้นมา

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11437 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1085809 (1.0 Mb)  TX bytes:420 (420.0 b)
          Interrupt:10 Base address:0xe000

eth0.1    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.2    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.3    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.4    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

กำหนด IP Address ให้กับ Interface ด้วยคำสั่ง ifconfig <interfacename> inet <ip address netmask <subnetmask> up

```
# ifconfig eth0.1 inet 192.168.1.254 netmask 255.255.255.0 up
# ifconfig eth0.2 inet 192.168.2.254 netmask 255.255.255.0 up
# ifconfig eth0.3 inet 192.168.3.254 netmask 255.255.255.0 up
# ifconfig eth0.4 inet 192.168.4.254 netmask 255.255.255.0 up
```


ตรวจสอบ IP Address ของ Interface ด้วยคำสั่ง ifconfig

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14325 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1342876 (1.2 Mb)  TX bytes:480 (480.0 b)
          Interrupt:10 Base address:0xe000

eth0.1    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:192.168.1.254  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.2    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:192.168.2.254  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.3    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:192.168.3.254  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth0.4    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:192.168.4.254  Bcast:192.168.4.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

ตรวจสอบ Routing Table ด้วยคำสั่ง netstat -rn

```
# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
195.168.1.0     0.0.0.0         255.255.255.0   U        40 0          0 eth1
192.168.4.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0.4
192.168.3.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0.3
192.168.2.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0.2
192.168.1.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0.1
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
```

กำหนดให้ทุกครั้งที่เครื่อง Server Boot ให้มีการเพิ่ม VLAN และกำหนด IP Address ให้กับ Interface ทำได้โดยเพิ่มข้อมูลต่อท้าย File /etc/rc.d/rc.local

```
/sbin/ifconfig eth0 inet 0.0.0.0 up
/sbin/vconfig add eth0 1
/sbin/vconfig add eth0 2
/sbin/vconfig add eth0 3
/sbin/vconfig add eth0 4
/sbin/ifconfig eth0.1 inet 192.168.1.254 netmask 255.255.255.0 up
/sbin/ifconfig eth0.2 inet 192.168.2.254 netmask 255.255.255.0 up
/sbin/ifconfig eth0.3 inet 192.168.3.254 netmask 255.255.255.0 up
/sbin/ifconfig eth0.4 inet 192.168.4.254 netmask 255.255.255.0 up
```

Config Catalyst 2924XL

เหมือนกับการ Config Catalyst 2924XL สำหรับ FreeBSD

IP Security โดย Freeswan บนระบบปฏิบัติการ Linux

จุดประสงค์การเรียนรู้

สามารถอธิบายขั้นตอนการทำงานและส่วนประกอบพื้นฐานของ IPSec ได้

สามารถติดตั้ง Package Freeswan เพื่อใช้งาน IPSec ได้

สามารถบอกตำแหน่งของ file ที่เก็บค่าเริ่มต้นการทำงานของ IPSec พร้อมหน้าที่การทำงานได้

สามารถสร้าง Private Key และPublic Key ที่ใช้สำหรับ Authentication ใน IPSec ได้

สามารถกำหนดค่าเริ่มต้น ติดตั้งและทดสอบการทำงานของการทำ IPSec แบบ Network to Network ได้

Web Site อ้างอิง

http://www.freeswan.org/download.html

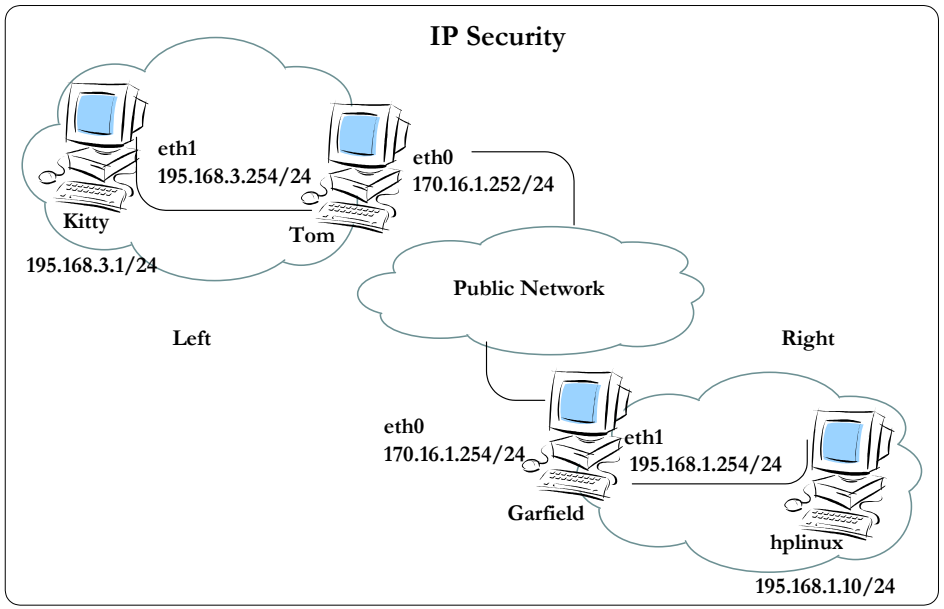
http://www.cipsa.org/members/routers.html

สำหรับ Mandrake Linux ติดตั้ง Package libcap และ freeswan

libpcap0-0.7.1-2mdk.i586.rpm

freeswan-1.98b-1mdk.i586.rpm

Network Model



การจัดการ Key ที่ใช้ในการ Authentication

/etc/freeswan/ipsec.secrets
เก็บ RSA Private key ใช้สำหรับ Authentication (กำหนด permission เป็น 600)

ที่เครื่อง Tom (left)
สร้าง hostkey (Private Key)

```
[root@tom root]# ipsec newhostkey --output privatekey
```

เรียบร้อยแล้ว move file privatekey ไปไว้ที่ /etc/freeswan/ipsec.secrets

```
[root@tom root]#mv privatekey /etc/ipsec.secrets
```

เก็บ public key ของ Host ฝั่งซ้ายลง File ชื่อ leftpkey

```
[root@tom root]#ipsec showhostkey --left  >leftpkey
```

```
[root@tom root]#cat leftpkey
# RSA 2192 bits   tom.info.com   Thu Jan  2 08:12:16 2003
leftrsasigkey=0sAQNkJIyHGm6PZSlClgQgre5wf1Kl59AD2kW3t/VXkHjQ56M
qRnKDka36jQ/MQZj9ioxb60PDyDdD22K305sr+SJjPuUaQVIHIKfljliXpqP09T
JD9HoUKo5nvPGIISVK/TPnYg1OzjgaQncDVTJRZuE7+x9Y05Qr6/I8tXexwvYPh
74Ah7UySkvcCUgKFK+xUL8k5jPnSnbBCvrBl1c0wg1p1TRPxz9POHQei+Nf2Qcn
kmPJcX/YphsJMhfhSg4wF22UBspUqeYwVAZXxGv8JYALFvBtwZpRkLpRkHpnBG+
DNcbweBJmD4/KaqlLft1rUmGH+r4DPLs3Y8Yi6FkVq0XHRNv75ijrVEegy+riJb
Xe0dGJ
```

เพิ่ม leftrsasigkey=.... (จาก file leftpkey) ลงใน file /etc/freeswan/ipsec.conf ในส่วน conn %default

```
conn %default
     keyingtries=0
     disablearrivalcheck=no
     authby=rsasig
     rightrsasigkey=0sAQN+l0OZA/j/O1LASOSLgwC0aFsz87IqZb (more)
     leftrsasigkey=0sAQNkJIyHGm6PZSlClgQgre5wf1Kl59AD2...(more)
```

เพิ่ม rightrsasigkey=.... (นำมาจาก file rightpkey ของเครื่อง Garfield)

ที่เครื่อง Garfield (right)

สร้าง hostkey (Private Key)

```
[root@garfield root]#ipsec newhostkey --output privatekey
```

เรียบร้อยแล้ว move file privatekey ไปไว้ที่ /etc/freeswan/ipsec.secrets

```
[root@garfield root]#mv privatekey /etc/ipsec.secrets
```

เก็บ public key ของ Host ฝั่งขวาลง File ชื่อ rightpkey

```
[root@garfield root]#ipsec showhostkey --right  >rightpkey
```

แสดงข้อมูลใน file rightpkey

```
[root@garfield root]#cat rightpkey
# RSA 2192 bits   garfield.info.com   Thu Jan  2 07:29:59 2003
rightrsasigkey=0sAQN+l0OZA/j/O1LASOSLgwC0aFsz87IqZbWdiJGr7fUvVw
m4HSe3xJsv1rPBEk/PmmjpN9lrDFSBDX76NcEaBLF54Xok5yycbN1nqzgbKGQg0
DlcXXPDyTXkeGCNFIJk2Qj5x5QBqdRrhzKAfgndWjExSfJuTd0BEHiQOMgp+ZEx
aAsfDHePvzm5G2W4DoMewZ89c95+5ndConCLedmJ/IGZ6SuFpahIDzDmd62dn49
GrFbWr9XmZiqwIXuqEDqI3QrlZQgwO5Z56IzFBNmflrqCbS/d9QyKiR1AY4azYp
jbRF5rk1XoP1OA91ENXm09Fczt8sM2ST5B+1SJP4Nb1sWDTbXy4eFAC+7BTU4PA
sw+DSIR
```

เพิ่ม rightrsasigkey=.... (จาก file rightpkey) ลงใน file /etc/freeswan/ipsec.conf ในส่วน conn %default

```
conn %default
     keyingtries=0
     disablearrivalcheck=no
     authby=rsasig
     rightrsasigkey=0sAQN+l0OZA/j/O1LASOSLgwC0aFsz87IqZb (more)
     leftrsasigkey=0sAQNkJIyHGm6PZSlClgQgre5wf1K159AD2...(more)
```

เพิ่ม leftrsasigkey=.... (นำมาจาก file leftpkey ของเครื่อง Tom)

กำหนดค่าเริ่มต้นการทำงาน

file /etc/freeswan/ipsec.conf เก็บรายละเอียดการทำงานของ IPSec

รูปแบบการต่อเชื่อม IPSec แบบ Tunnel

```
conn <connection name>
   left=<left ip>
   leftsubnet=<left subnet>
   leftnexthop=<left nexthop>
   right=<right ip>
   rightsubnet=<right subnet>
   rightnexthop=<right nexthop>
   auto=start
```

ถ้า left กับ right ต่อกันโดยตรงไม่ต้องกำหนดค่า leftnexthop (Default Gateway ของด้านซ้าย) และ rightnexthop (Default Gateway ของฝั่งขวา)

ทั้งที่ Garfield และ Tom เพิ่มที่ท้าย file /etc/freeswan/ipsec.conf

```
conn tom-to-garfield
    left=170.16.1.254
    leftsubnet=195.168.1.0/24
    right=170.16.1.252
    rightsubnet=195.168.3.0/24
    auto=start
```

สั่ง restart ipsec

```
#service ipsec restart
```

การทดสอบการทำงานของ IPSec

*(เมื่อสั่ง service network restart แล้วต้องสั่ง service ipsec restart ด้วย)

ที่เครื่อง Garfield

ตรวจสอบ Interface ด้วยคำสั่ง ifconfig ได้ Interface หลัก 2 Interface (eth0 และ eth1) และ Interface ipsec0

```
[root@garfield ]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:170.16.1.254  Bcast:170.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:726778 errors:4 dropped:198 overruns:0 frame:0
          TX packets:9923 errors:9 dropped:0 overruns:0 carrier:18
          collisions:300 txqueuelen:100
          RX bytes:67792464 (64.6 Mb)  TX bytes:7827182 (7.4 Mb)
          Interrupt:10 Base address:0xe000

eth1      Link encap:Ethernet  HWaddr 00:00:E8:1B:7A:BF
          inet addr:195.168.1.254  Bcast:195.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:736063 errors:4 dropped:0 overruns:0 frame:15
          TX packets:16864 errors:0 dropped:0 overruns:0 carrier:0
          collisions:534 txqueuelen:100
          RX bytes:74690003 (71.2 Mb)  TX bytes:1284448 (1.2 Mb)
          Interrupt:5 Base address:0x320

ipsec0    Link encap:Ethernet  HWaddr 00:02:44:0C:C1:1B
          inet addr:170.16.1.254  Mask:255.255.255.0
          UP RUNNING NOARP  MTU:16260  Metric:1
          RX packets:6725 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6762 errors:0 dropped:39 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:440038 (429.7 Kb)  TX bytes:7434836 (7.0 Mb)
```

ตรวจสอบ Routing Table ของเครื่องด้วยคำสั่ง route หรือ netstat -rn โดยเส้นทางที่จะไปยัง Network ทางด้านขวาจะส่งไปยัง 170.16.1.252 (เครื่อง tom)

```
[root@garfield ]# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
195.168.1.0     0.0.0.0         255.255.255.0   U        40 0          0 eth1
195.168.3.0     170.16.1.252    255.255.255.0   UG       40 0          0 ipsec0
170.16.1.0      0.0.0.0         255.255.255.0   U        40 0          0 eth0
170.16.1.0      0.0.0.0         255.255.255.0   U        40 0          0 ipsec0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
```


ตรวจสอบ function การทำงานของ ip forward จากคำสั่ง sysctl ต้องได้ผลลัพธ์เป็น 1

```
[root@garfield ]# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
```

ถ้ามีค่าเป็น 0 ให้ใช้คำสั่ง sysctl -w net.ipv4.ip_forward=1 และเข้าไปแก้ไขค่าใน file /etc/sysctl.conf

```
[root@garfield ]# sysctl -w net.ipv4.ip_forward=1
```

ตรวจสอบการทำงานของ iptables (firewall) ว่ายังไม่มีการทำงาน ถ้าทดสอบ ipsec เรียบร้อยแล้ว จึงกลับมากำหนดการทำงานของ ipsec ร่วมกันกับ firewall (เพื่อความสะดวกในทำงาน)

```
[root@garfield ]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

แสดงสถานะการทำงานของส่วนประกอบต่างๆ ของ IPSec จากคำสั่ง ipsec barf โดยผลลัพธ์ที่ได้จะเป็น logging ของขั้นตอนการทำงานต่างๆ

```
[root@garfield ]# ipsec barf
```

แสดงการทำงานของ IPSec ด้วยคำสั่ง ipsec look

```
[root@garfield demo]# ipsec look
garfield.info.com Wed Jan 15 11:32:30 ICT 2003
195.168.1.0/24     -> 195.168.3.0/24     => tun0x1008@170.16.1.252
esp0x638f301c@170.16.1.252  (641)
ipsec0->NULL mtu=16260(0)->0
life(c,s,h)=bytes(106440,0,0)addtime(11974,0,0)usetime(11718,0,0)packets(1124,0
,0) idle=1850
170.16.1.0      0.0.0.0         255.255.255.0   U        40 0          0 eth0
170.16.1.0      0.0.0.0         255.255.255.0   U        40 0          0 eth0
195.168.3.0     170.16.1.252    255.255.255.0   UG       40 0          0 ipsec0
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
```

ที่เครื่อง Tom

```
[root@tom root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:C8:7D:73:5B
          inet addr:170.16.1.252  Bcast:170.16.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:751252 errors:0 dropped:384 overruns:0 frame:14
          TX packets:8903 errors:0 dropped:0 overruns:0 carrier:0
          collisions:100 txqueuelen:100
          RX bytes:77209600 (73.6 Mb)  TX bytes:1133533 (1.0 Mb)
          Interrupt:9 Base address:0x280

eth1      Link encap:Ethernet  HWaddr 00:00:E8:1B:78:D5
          inet addr:195.168.3.254  Bcast:195.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:746541 errors:2 dropped:0 overruns:0 frame:9
          TX packets:2337 errors:0 dropped:0 overruns:0 carrier:0
          collisions:11 txqueuelen:100
          RX bytes:69893706 (66.6 Mb)  TX bytes:322583 (315.0 Kb)
          Interrupt:5 Base address:0x320

ipsec0    Link encap:Ethernet  HWaddr 00:80:C8:7D:73:5B
          inet addr:170.16.1.252  Mask:255.255.255.0
          UP RUNNING NOARP  MTU:16260  Metric:1
          RX packets:6765 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6785 errors:0 dropped:24 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:6984067 (6.6 Mb)  TX bytes:895806 (874.8 Kb)
```

ตรวจสอบ Routing Table ของเครื่องด้วยคำสั่ง route หรือ netstat -rn โดยเส้นทางที่จะไปยัง Network ทางด้านขวาจะส่งไปยัง 170.16.1.254 (เครื่อง garfield)

```
[root@tom root]# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
195.168.1.0     170.16.1.254    255.255.255.0   UG        40 0          0
ipsec0
195.168.3.0     0.0.0.0         255.255.255.0   U         40 0          0 eth1
170.16.1.0      0.0.0.0         255.255.255.0   U         40 0          0 eth0
170.16.1.0      0.0.0.0         255.255.255.0   U         40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         40 0          0 lo
```

ตรวจสอบ function การทำงานของ ip forward จากคำสั่ง sysctl ต้องได้ผลลัพธ์เป็น 1

```
[root@tom root]# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
```


ถ้ามีค่าเป็น 0 ให้ใช้คำสั่ง sysctl -w net.ipv4.ip_forward=1 และเข้าไปแก้ไขค่าใน file /etc/sysctl.conf

```
[root@tom root ]# sysctl -w net.ipv4.ip_forward=1
```

แสดงสถานะการทำงานของ iptables

```
[root@tom root]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

แสดงสถานะการทำงานของส่วนประกอบต่างๆ ของ IPSec จากคำสั่ง ipsec barf โดยผลลัพธ์ที่ได้จะเป็น logging ของขั้นตอนการทำงานต่างๆ

```
[root@tom root]# ipsec barf
```

แสดงการทำงานของ IPSec ด้วยคำสั่ง ipsec look

```
[root@tom root]# ipsec look
tom.info.com Wed Jan 15 11:41:12 ICT 2003
195.168.3.0/24     -> 195.168.1.0/24     => tun0x1008@170.16.1.254
esp0xa4ef23a3@170.16.1.254  (1124)
ipsec0->NULL mtu=16260(0)->0
esp0x638f301c@170.16.1.252 ESP_3DES_HMAC_MD5: dir=in  src=170.16.1.254
iv_bits=64bits iv=0x0763f3abb6df2624 ooowin=64 seq=641 bit=0xffffffffffffffff
alen=128 aklen=128 eklen=192
170.16.1.0      0.0.0.0         255.255.255.0   U         40 0          0 eth0
170.16.1.0      0.0.0.0         255.255.255.0   U         40 0          0 eth0
195.168.1.0     170.16.1.254    255.255.255.0   UG        40 0          0
ipsec0
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
```

ที่เครื่อง Kitty

กำหนด IP Address 195.168.3.1/24 Default Gateway 195.168.3.254

```
[root@kitty root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:E8:A7:02:96
          inet addr:195.168.3.1  Bcast:195.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9347668 errors:0 dropped:0 overruns:0 frame:45
          TX packets:224747 errors:2 dropped:0 overruns:0 carrier:2
          collisions:334 txqueuelen:100
          RX bytes:1252156540 (1194.1 Mb)  TX bytes:19260334 (18.3 Mb)
          Interrupt:10 Base address:0x280
```

```
[root@kitty root]# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
195.168.3.0     0.0.0.0         255.255.255.0   U        40 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        40 0          0 lo
0.0.0.0         195.168.3.254   0.0.0.0         UG       40 0          0 eth0
```

ทดลอง ping ไปยัง 195.168.1.254 (Interface ด้านในของเครื่อง garfield)

```
[root@kitty root]# ping 195.168.1.254
PING 195.168.1.254 (195.168.1.254) from 195.168.3.1 : 56(84) bytes of data.
64 bytes from 195.168.1.254: icmp_seq=1 ttl=63 time=6.19 ms
64 bytes from 195.168.1.254: icmp_seq=2 ttl=63 time=3.36 ms
64 bytes from 195.168.1.254: icmp_seq=3 ttl=63 time=3.40 ms
```

ping ไปยังเครื่อง hplinux (195.168.1.10)

```
[root@kitty root]# ping 195.168.1.10
PING 195.168.1.10 (195.168.1.10) from 195.168.3.1 : 56(84) bytes of data.
64 bytes from 195.168.1.10: icmp_seq=1 ttl=62 time=4.59 ms
64 bytes from 195.168.1.10: icmp_seq=2 ttl=62 time=3.78 ms
64 bytes from 195.168.1.10: icmp_seq=3 ttl=62 time=3.80 ms
```

ไม่สามารถ ping 170.16.1.254 ได้เพราะการทำงานของ tunnel จะส่งข้ามไปยัง network 195.168.1.0
จากเครื่อง hplinux (195.168.1.10) ping ไปยังเครื่อง kitty (195.168.3.1) ได้ผลเช่นเดียวกัน