ปรับแต่งระบบปฏิบัติการ FreeBSD ด้วยการ Compile Kernel

เมื่อเราติดตั้งระบบปฏิบัติการ FreeBSD เสร็จ โดยปกติแล้วเราจะได้ GENERIC kernel ซึ่ง FreeBSD เตรียมไว้ให้ โดย GENERIC Kernel นั้นจะรู้จัก Hardware จำนวนมาก ทำให้ Server ของเราต้องตรวจสอบ Hardware ว่าเครื่อง Server ของเรามีอะไรบ้าง เสียเวลาในการ boot ระบบ อีกทั้งต้องใช้หน่วยความจำหลักในการทำงานดังกล่าวพอสมควร เราสามารถทำการ Compile Kernel ใหม่ให้เหมาะสมกับ Hardware ของเราได้

ในการ Compile Kernel ในบทนี้จะเป็นเพียงการเพิ่มความสามารถให้กับระบบปฏิบัติการ FreeBSD ให้สามารถทำงานเป็น Gateway ให้กับ Network ย่อยของเรานั่นคือให้ FreeBSD server ของเราทำหน้าที่เป็น NAT (Network Address Translation) และ Firewall

ขั้นตอนการ Compile Kernel

1. เมื่อตอนติดตั้งเราได้เลือก src ไว้แล้ว ให้เราเข้าไปที่เก็บ GENERIC Kernel แล้วทำการ copy GENERIC Kernel เป็น ชื่อ Kernel ของเรา เช่น GATEWAY , PROXY ฯลฯ ในที่นี้เราจะใช้ชื่อ PROXY

```
#cd /usr/src/sys/i386/conf
#cp GERNERIC PROXY
```

2. ตรวจสอบ CPU ของเราว่า เป็น Class ใด โดยใช้คำสั่ง

```
#dmesg | grep CPU
```

จะได้ผลลัพธ์ดังภาพ นั่นคือ 686-class CPU

```
proxy# dmesg ¦ grep CPU
CPU: Intel(R) Celeron(R) CPU 2.80GHz (2809.11-MHz 686-class CPU)
cpu0: <ACPI CPU> on acpi0
acpi_throttle0: <ACPI CPU Throttling> on cpu0
CPU: Intel(R) Celeron(R) CPU 2.80GHz (2809.71-MHz 686-class CPU)
cpu0: <ACPI CPU> on acpi0
acpi_throttle0: <ACPI CPU Throttling> on cpu0
proxy#
```

3. เมื่อตรวจสอบ CPU แล้วให้เราทำการแก้ไขไฟล์ Kernel ของเรา

```
#ee  PROXY
```

4.ในไฟล์ PROXY ให้แก้ไข บรรทัดต่อไปนี้

ก่อนแก้ไข		หลังแก้ไข	
machine	i386	machine	i386
cpu	I486_CPU	**#**cpu	I486_CPU
cpu	I586_CPU	**#**cpu	I586_CPU
cpu	I686_CPU	cpu	I686_CPU
ident	GENERIC	ident	**PROXY**

5. ในไฟล์เดียวกัน ให้เราทำการเพิ่ม options ต่าง ๆ ต่อไปนี้ ก่อน options เดิมที่มีอยู่แล้ว เพื่อให้ Kernel ของเรา ทำงานเป็น NAT และ Firewall รวมทั้ง เพิ่ม options QUOTA เพื่อให้เราสามารถจัดการจำกัดขนาดพื้นที่ให้ user ใน server ใช้งานตามโควตา ในภายหลัง หรือ เพิ่ม

options DUMMYNET

options HZ=1000

เพื่อให้เราสามารถใช้ Dummynet ในการจัดการกำหนด bandwidth ให้กับเครือข่ายต่าง ๆ ที่เชื่อมต่อได้

```
options         IPFIREWALL
options         IPFIREWALL_FORWARD
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPFIREWALL_VERBOSE
options         IPFIREWALL_VERBOSE_LIMIT=120
options         IPDIVERT
```

6. ให้ทำการ Save ไฟล์โดยการกด Ctl + C แล้วพิมพ์ exit เพื่อออกมาที่ prompt ของ ระบบหลังจากนั้นให้ทำการ config Kernel ที่เราได้แก้ไขแล้ว โดยใช้คำสั่งดังนี้

```
#config   PROXY
```

จะได้ผลดังนี้

```
Kernel build directory is ../compile/PROXY
Don't forget to do "make cleandepend; make depend"
```

7. ให้เราเข้าไปทำการ compile kernel และติดตั้ง โดยใช้คำสั่ง

```
#cd   ../compile/PROXY
#make cleandepend ; make depend
รอสักครู่.....
#make ; make install clean
```

8. ให้เราทำการ reboot ระบบใหม่

```
#reboot
หรืออาจจะใช้คำสั่งต่อไปนี้แทนก็ได้ สังเกต –r  =  reboot , -p = power off
#shutdown –r now
```

9. ตรวจสอบระบบว่า boot ด้วย Kernel ใหม่ที่ชื่อ PROXY หรือไม่

```
#uname –a
        ต้องได้ผลคล้าย ๆ แบบนี้
FreeBSD proxy.intra.net 6.3-RELEASE FreeBSD 6.3-RELEASE #0: Thu Apr 25 14:04:19 ICT
2008       root@proxy.intra.net:/usr/src/sys/i386/compile/PROXY  i386
```

ขณะนี้เราได้ Kernel ใหม่ที่รองรับการทำ NAT และ Firewall แล้ว แต่ Server จะยังไม่ทำงานเป็น NAT และ Firewall จนกว่าเราจะ config ให้ server ทำงานดังนี้

ให้เราทำการ config ค่าต่าง ๆ เหล่านี้ ในไฟล์ /etc/rc.conf เพิ่มเติม

```
#ee /etc/rc.conf
```

เพิ่มบรรทัดต่อไปนี้

```
firewall_enable="YES"
firewall_type="OPEN"
firewall_quite="YES"
natd_enable="YES"
natd_interface="rl0"
natd_flags="-s -u -m"
```

หมายเหตุ : rl0 คือ LAN Card ที่ต่อกับ internet

โดยต้องมีบรรทัดต่อไปนี้ อยู่ด้านบนในไฟล์ /etc/rc.conf หากไม่มีให้พิมพ์เพิ่มเข้าไป

```
gateway_enable="YES"
```

เพื่อให้ server ของเรา ทำหน้าที่เป็น gateway ของ network ได้อย่างสมบูรณ์

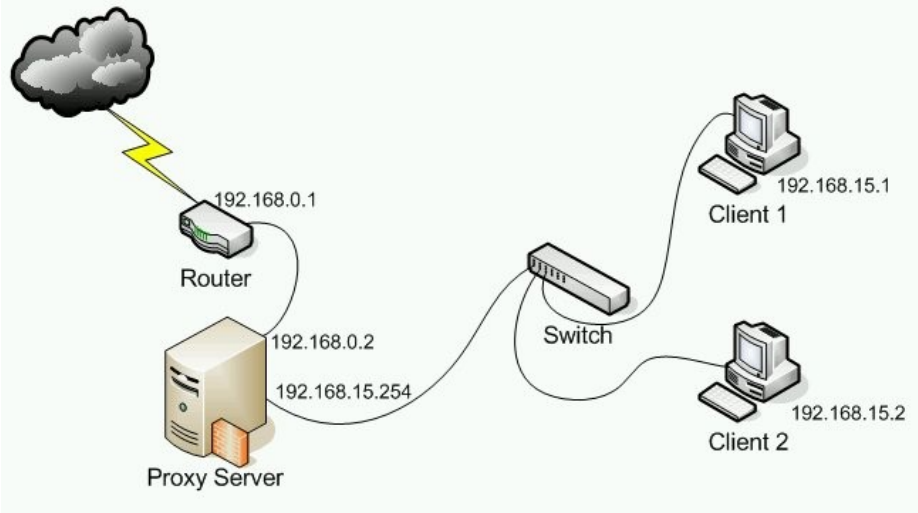
ให้ทำการ reboot server อีกครั้ง แล้วให้ลองใช้คำสั่ง

```
#ipfw show
```

เพื่อตรวจสอบการทำ NAT ซึ่ง Firewall จะยอมให้ทุก package ที่ เข้า - ออก เครื่องลูกข่าย ผ่านทาง NAT Server ตัวนี้ได้ ซึ่งจะได้ผลคล้ายแบบนี้

```
00050   505   57856 divert 8668 ip from any to any via rl0
00100    12    1008 allow ip from any to any via lo0
00200     0       0 deny ip from any to 127.0.0.0/8
00300     0       0 deny ip from 127.0.0.0/8 to any
65000   639   77505 allow ip from any to any
65535     0       0 allow ip from any to any
```

เพื่อให้เข้าใจในการทำ NAT นั้นจะขออธิบายเพิ่มเติมดังนี้



จากรูปแสดงเครือข่ายด้านบน ในไฟล์ /etc/rc.conf จะมีลักษณะคล้ายดังนี้

```
gateway_enable="YES"
hostname="proxy.intra.net"
ifconfig_rl0=”inet 192.168.0.2  netmask 255.255.255.0”
ifconfig_rl1=”inet 192.168.15.254  netmask 255.255.255.0”
```

โดย rl0 คือ LAN Card ใบแรก ที่ใช้ต่อไปยังเครือข่าย internet ส่วน rl1 นั้น จะเป็น LAN Card ที่ต่อกับเครือข่ายภายในของเรา ทั้งนี้หาก Card LAN ของเราไม่ใช่ยี่ห้อ Realtek จะไม่ใช่ชื่อ rl โดยอาจจะเป็นชื่ออื่น ให้ลองใช้คำสั่งต่อไปนี้ ตรวจสอบดู

#ifconfig จะได้ผลคล้าย ๆ แบบนี้

```
demo# ifconfig
lnc0: flags=108843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 172.28.5.11 netmask 0xffff0000 broadcast 172.28.255.255
        inet6 fe80::20c:29ff:fed7:a0d8%lnc0 prefixlen 64 scopeid 0x1
        ether 00:0c:29:d7:a0:d8
lnc1: flags=108843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.100.254 netmask 0xffffff00 broadcast 192.168.100.255
        inet6 fe80::20c:29ff:fed7:a0e2%lnc1 prefixlen 64 scopeid 0x2
        ether 00:0c:29:d7:a0:e2
plip0: flags=108810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
```

ตัวอย่างรหัส Card LAN ที่พบได้บ่อย เช่น

```
de      # DEC/Intel DC21x4x (“Tulip”)
em      # Intel PRO/1000 adapter Gigabit Ethernet Card (“Wiseman”)
txp     # 3Com 3cR990 (“Typhoon”)
vx      # 3Com 3c590, 3c595 (“Vortex”)
dc      # DEC/Intel 21143 and various workalikes
fxp     # Intel EtherExpress PRO/100B (82557, 82558)
pcn     # AMD Am79C97x PCI 10/100 NICs
rl      # RealTek 8129/8139
sf      # Adaptec AIC-6915 (“Starfire”)
sis     # Silicon Integrated Systems SiS 900/SiS 7016
ste     # Sundance ST201 (D-Link DFE-550TX)
tl      # Texas Instruments ThunderLAN
tx      # SMC EtherPower II (83c170 “EPIC”)
vr      # VIA Rhine, Rhine II
wb      # Winbond W89C840F
xl      # 3Com 3c90x (“Boomerang”,”Cyclone”)
bge     # Broadcom BCM570x (“Tigon III”)
```

Proxy Server หรือ Cache Server

Proxy server คือ server ที่กั้นอยู่ตรงกลางระหว่าง workstation กับ Internet ทำหน้าที่เป็น web caching :ซึ่ง proxy server อาจจะเป็นตัวเดียวกับ gateway server ที่ทำหน้าที่แยกเครือข่ายภายในองค์กรจากเครือข่ายภายนอก

โปรแกรมที่ใช้ในการทำ caching คือโปรแกรม Squid (http://www.squid-cache.org)

การติดตั้ง Squid-Cache

1. ให้ทำการ copy ไฟล์ squid-2.6.STABLE18.tar.gz จากแผ่น CD ไปไว้ยัง Directory /tmp

```
#cd   /tmp
#mount  /cdrom
#cp   /cdrom/proxy/squid-2.6.STABLE18.tar.gz   ./
#umount   /cdrom
```

2. ให้แตกไฟล์ที่ copy มาจาก CD และทำการติดตั้ง ดังนี้

```
#tar   -zxvf   squid-2.6.STABLE18.tar.gz
#cd squid-2.6.STABLE18
#./configure   --prefix=/usr/local/squid   --enable-removal-policies=heap   --enable-delay-pools
 --enable-snmp   --enable-arp-acl   --enable-basic-auth-helpers="NCSA"
#make all ; make install
#rehash
```

3. สร้าง logs ไฟล์เพื่อใช้ในการเก็บ log การเข้าดูเวบของเครื่องลูกข่าย

```
#cd /usr/local/squid/var/logs/
#touch access.log
#touch store.log
#touch cache.log
#cd /usr/local/squid/var/
#chmod   -R 777   logs/
```

4. สร้าง Directory ไว้เก็บเวบต่าง ๆ ที่เครื่องลูกข่ายเข้าไปดูมา

```
#cd   /usr/local/squid/var/
#mkdir   cache
#chmod   -R   777   cache
```

5. แก้ไข config ไฟล์ของโปรแกรม squid

```
***** ให้คัดลอกไฟล์ squid.conf มาจากแผ่น cdrom (/cdrom/proxy/config/squid.conf20080404)
#cd   /usr/local/squid/etc/
#ee squid.conf
```

โดยให้แก้ไขบรรทัดต่อไปนี้ (ตำแหน่งต่าง ๆ ดูจากไฟล์ตัวอย่าง)

บรรทัด http_port 3128 ให้แกไขเป็น

http_port 8080 (ประมาณบรรทัดที่ 998)

บรรทัด # cache_dir ufs /usr/local/squid/var/cache 100 16 256 ให้ copy ลงมา อีก 1 บรรทัด แล้วจึงแก้เป็น

cache_dir ufs /usr/local/squid/var/cache 500 16 256 (ประมาณบรรทัดที่ 1903)

เพิ่มบรรทัดต่อไปนี้เพื่อให้ cache มีประสิทธิภาพมากขึ้น

memory_replacement_policy heap GDSF (ประมาณบรรทัดที่ 1711)

cache_replacement_policy heap GDSF (ประมาณบรรทัดที่ 1754)

เพิ่มบรรทัดต่อไปนี้เพื่อให้ ระบบถาม Username ทุกครั้งที่เข้าใช้อินเตอร์เน็ต (ประมาณบรรทัดที่ 280 เป็นต้นไป)

```
auth_param basic program /usr/local/squid/libexec/ncsa_auth /usr/local/squid/etc/passwd
auth_param basic children 5
auth_param basic realm Some School proxy-caching web server
auth_param basic credentialsttl 1 hours
auth_param basic casesensitive off
```

เพิ่มบรรทัดต่อไปนี้เพื่อให้ proxy ถาม username password จากเครื่องลูกข่ายที่เข้ามาจากเครือข่ายที่กำหนด (ประมาณบรรทัดที่ 608)

```
acl auth_users proxy_auth REQUIRED
acl schoolnet src 192.168.15.0/255.255.255.0

acl bansites url_regex "/usr/local/squid/etc/bansites.txt"
http_access deny bansites
deny_info http://192.168.15.254:81/bansites.html bansites

acl block url_regex "/usr/local/squid/etc/block.txt"
http_access deny block all

acl worktime time SMTWHFA 08:30-12:00 13:00-16:45
acl LTA url_regex "/usr/local/squid/etc/lta.txt"
http_access allow !worktime LTA
http_access deny worktime LTA
```

ส่วนที่เพิ่มเพื่อให้ระบบยอมให้ user ที่ใส่ password ถูกต้องเข้าใช้งานอินเตอร์เน็ตได้

```
http_access allow auth_users
http_access allow schoolnet

##delay-pools##
acl my_local url_regex -i 192.168.15.
acl my_unlimit time SMTWHFA 08:00-18:00
acl my_download url_regex "/usr/local/squid/etc/delaylists.txt"
delay_pools 2
delay_class 1 2
delay_parameters 1 -1/-1 -1/-1
delay_access 1 allow my_local
delay_access 1 allow !my_unlimit
delay_access 1 deny my_unlimit
delay_class 2 2
delay_parameters 2 512000/512000 20000/20000
delay_access 2 allow my_download
```

ที่บรรทัด ประมาณ 3639 ให้ใส่เครื่องหมาย # หน้าบรรทัด ที่มีคำว่า

```
snmp_access allow snmppublic mrtg
```

ที่บรรทัด ประมาณ 1693 เปลี่ยน cache_mem 500 MB เป็น

```
cache_mem 8 MB
```

หลังจากนั้นให้ทำการ save ไฟล์ เพื่อทำงานต่อ

6. เพิ่มความสะดวกในการเรียกใช้ squid จาก directory ใดก็ได้

```
#cd /usr/sbin/
#ln -s /usr/local/squid/sbin/squid squid
#rehash
```

7. สร้างโครงสร้างการเก็บข้อมูลของ squid ใน /usr/local/squid/var/log/cache ด้วยคำสั่ง

```
#squid –z
```

8. ให้สร้างไฟล์ที่ชื่อว่า squid.sh ไว้ใน /root เพื่อใช้ในการ start / stop squid

```
#cd /root/
#ee squid.sh
```

ให้พิมพ์ข้อความเหล่านี้ลงไปในไฟล์ (เตรียมไว้ให้แล้วใน cdrom /cdrom/proxy/config/squid.sh)

```
#!/bin/sh
case "$1" in
start)
if [ -f /usr/local/squid/bin/RunCache ]; then
        echo -n 'starting Squid ...'
        (/usr/local/squid/bin/RunCache &)
fi
;;
stop)
kill -9 `ps ax | grep RunCache | awk '{print $1}'`;
/usr/local/squid/sbin/squid -k shutdown;
;;
*)
echo "squid.sh stop|start"
;;
esac
exit 0
```

เมื่อพิมพ์เสร็จแล้วให้ทำการ save และเปลี่ยน permission ไฟล์ เพื่อให้ shell script ตัวนี้ ทำงานได้

```
#chmod 755 squid.sh
```

ในการเรียกใช้งาน ให้ทำดังนี้

```
#cd /root/
#./squid.sh   stop
        เพื่อ stop squid process
#./squid.sh   start
        เพื่อ start squid process
```

9. สั่งให้ squid ทำงาน

```
#cd
#squid.sh   start
```

10. ให้เพิ่ม กฎ ของ firewall ต่อไปนี้ เพื่อให้ลูกข่ายสามารถใช้ internet ได้

```
#ipfw add 1301 fwd 192.168.15.254,8080 tcp from 192.168.15.0/24 to any 80,ftp in via rl0
```

11. ลองใช้เครื่องลูกข่ายเข้าอินเตอร์เน็ตดู แล้วใช้คำสั่งต่อไปนี้ตรวจสอบการทำงานของ Squid

```
#tail -f /usr/local/squid/var/logs/access.log
```

12. หากมีการแก้ไขค่า config ของ squid ให้ใช้คำสั่งต่อไปนี้เพื่อให้ squid ทำงานตามการ config ค่าใหม่

```
#/usr/local/squid/sbin/squid –k reconfig
```

หาก config ค่าตามตัวอย่างต้องสร้างไฟล์เพิ่ม 5 ไฟล์ คือ

1. /usr/local/squid/etc/passwd ต้องติดตั้ง โปรแกรม Apache เพื่อให้สามารถใช้คำสั่งสร้างชื่อผู้ใช้งานอินเตอร์พร้อม password

   #htpasswd –cb /usr/local/squid/etc/passwd username1 password1 หรือ

   #htpasswd –b /usr/local/squid/etc/passwd username2 password2
2. /usr/local/squid/etc/bansites.txt เก็บรายชื่อเวบที่เราไม่ต้องการให้ลูกข่ายเข้าดู
3. /usr/local/squid/etc/block.txt เก็บรายชื่อเวบที่เราป้องกันการเข้าถึง
4. /usr/local/squid/etc/lta.txt เก็บรายชื่อเวบที่เราอนุญาตให้เข้าเป็นเวลา
5. /usr/local/squid/etc/delaylists.txt เก็บรายชื่อเวบที่เราต้องการกำหนดความเร็วในการเข้าดูหรือความเร็วในการดาวน์โหลด

สร้างไฟล์ /root/rotate_and_keep_proxy_log เพื่อใช้เก็บ log ที่เราสามารถอ่านวันเวลาได้

```
#!/bin/sh
day=`date '+%Y%m%d'`
if [ -f /root/logs/access.log.cache.${day} ]; then
      exit 0
fi
squid -k rotate
cat /usr/local/squid/var/logs/access.log.0 | awk '{print $1 " " $3 " " $8 " " $6 " " $7}' | \
perl -pe 's/^\d+\.\d+/localtime($&)/e;' > /root/logs/access.log.cache.${day}
gzip /root/logs/access.log.cache.${day}
```

สร้าง Directory /root/logs เพื่อเก็บ log ไฟล์แยกเป็นรายวัน

```
#mkdir /root/logs
```

ในไฟล์ /usr/local/squid/etc/squid.conf เพิ่มบรรทัด ต่อไปนี้เข้าไป

```
logfile_rotate 1
```

ในแฟ้ม /etc/crontab เพิ่มบรรทัดต่อไปนี้เข้าไปด้านล่างสุดต่อจากของเดิมที่มีอยู่

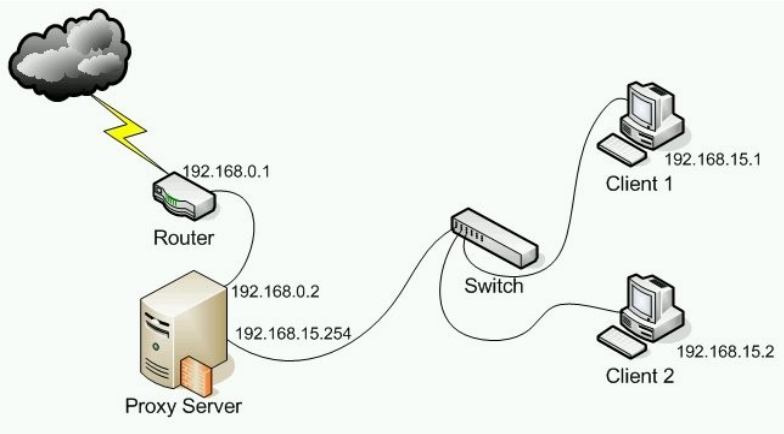
```
0  0  *  *  *  root  /root/rotate_and_keep_proxy_log
```

เพื่อให้ squid เริ่มสร้าง log ไฟล์ใหม่เป็นรายวัน

***** หากต้องการสร้างรายงานแบบดูผ่านเวบได้ ให้ดูเอกสารการติดตั้งโปรแกรม Sarg

DHCP Server

Dynamic Host Configuration Protocol เป็น protocol ที่ใช้ในการกำหนด IP address ให้กับอุปกรณ์หรือเครื่องคอมพิวเตอร์ในเครือข่ายแบบ dynamic ซึ่งการกำหนด IP Address แบบ dynamic นี้ อุปกรณ์หรือเครื่องคอมพิวเตอร์จะได้รับ IP address ที่แตกต่างกันไปในแต่ละครั้งที่เชื่อมต่อมายังเครือข่าย แต่ DHCP ก็ยังสามารถกำหนด IP Address แบบ static IP address ได้เช่นกัน



ขั้นตอนการติดตั้ง

ในการติดตั้งนี้ เราจะทำการติดตั้งผ่าน ports ของ FreeBSD โดยถ้าหากยังไม่เคย copy ไฟล์ f63distfiles.tar.gz จากแผ่น CD ก็ให้ทำการ copy ไฟล์ f63distfiles.tar.gz จากแผ่น CD ที่เตรียมให้ ไปเก็บไว้ที่ /usr/ports/distfiles/ และเมื่อ copy เรียบร้อยแล้ว จึงค่อยแตกไฟล์ ดังกล่าวออกมา ดังนี้

```
#cd  /usr/ports/distfiles/
#mount  /cdrom
#cp  /cdrom/f63distfiles.tar.gz  ./
#umount  /cdrom
#tar  -zxvf  f63distfiles.tar.gz
#rm  f63distfiles.tar.gz
```

1. ให้เข้าไปที่ ports ของ DHCP แล้วทำการติดตั้ง

```
#cd  /usr/ports/net/isc-dhcp3-server/
#make  ;  make install
#rehash
```

ในการติดตั้งเมื่อเกิด Dialog ถาม การติดตั้ง ให้กด tab มาที่ OK แล้วจึง Enter เพื่อทำงานต่อ

2. เมื่อติดตั้งเสร็จให้แก้ไขไฟล์ /usr/local/etc/dhcpd.conf

```
#ee /usr/local/etc/dhcpd.conf
```

3. เพิ่มเติมข้อความต่อไปนี้ในไฟล์ dhcpd.conf (โดยอ้างอิงการเชื่อมต่อจากรูปเครือข่ายด้านบน)

```
option  domain-name     "intra.net";
option  domain-name-servers       192.168.15.254;
default-lease-time          86400;
max-lease-time              172800;
authoritative;
ddns-update-style           ad-hoc;
log-facility                local7;
subnet  192.168.15.0  netmask  255.255.255.0{
        range  192.168.15.100    192.168.15.200;
        option  subnet-mask       255.255.255.0;
        option  broadcast-address           192.168.15.255;
        option  routers   192.168.15.254;
}
```

4. หากต้องการ Fix IP address ให้กับคอมพิวเตอร์ลูกข่ายเครื่องใดเราต้องทราบ MAC Address ของ LAN Card ของคอมพิวเตอร์เครื่องนั้น ให้เราเพิ่มข้อความคล้ายตัวอย่างต่อไปนี้เข้าไปในไฟล์ dhcpd.conf ด้วย เช่น

```
host r31119c01{
         hardware ethernet 00:14:85:52:f3:d2;
         fixed-address 192.168.15.10;
}
```

5. ให้ DHCP server ทำงาน

```
#dhcpd -cf /usr/local/etc/dhcpd.conf -lf /var/db/dhcpd/dhcpd.leases
```

6. ให้เพิ่มบรรทัดต่อไปนี้ในไฟล์ /etc/rc.conf เพื่อให้ DHCP ทำงานทุกครั้งที่ boot เครื่องใหม่

```
dhcpd_enable="YES"          # dhcpd enabled?
dhcpd_flags="-q"            # command option(s)
dhcpd_conf="/usr/local/etc/dhcpd.conf"   # configuration file
dhcpd_ifaces="rl1"          # ethernet interface(s)
dhcpd_withumask="022"       # file creation mask
```

7. หากตั้งค่าต่างๆ ตามตัวอย่างนี้ หากเครื่องลูกข่ายที่เชื่อมต่อกับ switch ก็จะได้รับ IP Address โดยอัตโนมัติ รวมถึงค่า subnet mask , DNS เพื่อใช้ในการเชื่อมต่ออินเตอร์เน็ตต่อไป