Ahtapot
Saldırı Tespit ve Önleme Sistemi
Ansible Kurulum Test Dokümanı

1- Suricata Kurulum

Ayar dosyası içerisinde aşağıdaki değişkenler ayarlanır. suricata_mode: **ids** olarak ayarlanır.
IDS modunda suricata'nın dinleyeceği her bir interface **suricata_pcap_interfaces:** bölümü altında ayrı ayrı tanımlanır.

IPS ayarları için sadece suricata_mode: **ips** olarak ayarlanır. Interface tanımlamaya gerek yoktur.

IDS Yapılandırması

Ayar Dosyası: /etc/ansible/roles/suricata/vars/main.yml

Parametreler:

```
suricata_mode: ids #ids|ips
suricata_pcap_interfaces:
        - int: enp0s3
          buffer_size: 16777216
          bpf_filter: "tcp and port 25"
          checksum_checks: auto
          threads: 16
          promisc: no
          snaplen: 1518

        - int: enp0s8
          buffer_size: 16777216
          bpf_filter: "tcp and port 25"
          checksum_checks: auto
          threads: 16
          promisc: no
          snaplen: 1518
```

Komut: ansible-playbook /etc/ansible/playbooks/suricata.yml

Sonuç: Servis durumu kontrol edilir ve çalışır durumda olduğu görülür.

Komut : systemctl status suricata

Sonuç : Active: active (running) since Wed 2018-06-06 13:00:12 +03; 20s ago

IPS yapılandırması

IPS için Suricata'nın NFQUEUE modu kullanılır. NFQ iptables'in bir özelliğidir. Bu özellik sayesinde trafik Queue'lara yönlendirilir. Suricata bu Queue'da trafiği **inline** analiz eder ve aktif olan imzalarda **drop** olanları düşürür, **alert** olanlar için uyarı üretir ve trafiği iptables'a geri verir.

Birden fazla Queue suricata daemon'na yönlendirilebilir. Queue sayısı CPU core sayısına eş olacak veya bir eksiği olacak şekilde ayarlanması önerilir.

NPROC=$(/usr/bin/nproc)
NPROC=$(($NPROC -1)) && echo $NPROC

NFQUEUE iptables kurallarının yazımı **FWBuilder** üzerinden yapılır. **NAT/ROUTER** veya **Bridge** mod olarak konumlanmış güvenlik duvarlarında farklı kurallar yazılır.

Firewall kuralları ayarlandıktan sonra Suricata'nın NFQUEUE modunda çalışması için servis ayarları yapılır.

Ayar Dosyası: /etc/ansible/roles/suricata/vars/main.yml
Parametreler:
suricata_mode: ips #ids|ips

Komut: ansible-playbook /etc/ansible/playbooks/suricata.yml
Sonuç: Servis durumu kontrol edilir ve çalışır durumda olduğu görülür.
Komut : systemctl status suricata
Sonuç : Active: active (running) since Wed 2018-06-06 13:00:12 +03; 20s ago

FWBuilder üzerinde aşağıdaki yol takip edilerek kurallar eklenir ve politikalar ilgili sistemlere gönderilir.

NAT/ROUTER Mod IPS

1- Firewalls > Firwall_Obje > Policy > Çift Klik > Top Rule Set [ ] işareti kaldırılır.
2- Firewalls > Firwall_Obje > Sağ Klik > New Policy Rule Set denilerek yeni policy oluşturulur.
Name: 00_SuricataNFQ
[x] Top Rule Set işaretlenir.
3- Eklenen yeni Policy tablosuna 1 kural eklenir ve **Options** kısmında **logging off** seçilir.
4- Eklenen kuralın **Rule Options** kısmında aşağıdaki işlemler yapılır.
a- Assume firewall is part of "any" for this rule only: **off**
b- Stateless Rule [ ] işareti kaldırılır.
5- Kuralın ACTION kısmında Custom seçilir ve aşağıdaki değer yazılır.
#nproc komut ile cpu core sayısı belirlenir ve aşağıdaki kuraldaki uygun alana yazılır.
/usr/bin/nproc
Cpu Core Sayısı: 4

-j NFQUEUE --queue-balance 1:4 --queue-bypass

#Eğer 1 tane cpu core var ise kural'ın ACTION kısmı aşağıdaki gibi yazılır.

-j NFQUEUE --queue-num 1 --queue-bypass

Not: **queue-bypass** parametresi ile suricata servisi çalışmıyor ise trafiği alttaki kurallardan işletmeye devam eder.
Suricata NFQUEUE Yapılandırması

Bridge Mod IPS

Bridge mod IPS ayarları için öncelikle Bridge interface aşağıdaki yapılandırma örneğindeki gibi yapılandırılır.

Not: Bridge yapılandırması için bridge-utils ve ethtool paketlerinin sistemde kurulu olması gerekmektedir.
Hangi iki fiziksel interface bridge interface'in üyesi olacak ise bridge_ports parametresi ile tanımlanır. Ethtool ile özel ayarlar yapılması istenen fiziksel interface'ler tanımlanır.

Ayar Dosyası: /etc/network/interfaces

```
auto br0
iface br0 inet manual
        bridge_ports eth1 eth2
        bridge_stp on
        bridge_fd 5

        #Advanced Inteface Settings
        #post-up /sbin/ethtool -K eth1 tx off rx off sg off gso off gro off 1>/dev/null
        #post-up /sbin/ethtool -K eth2 tx off rx off sg off gso off gro off  1>/dev/null


        #SYSCTL Ayarları yapılır.
        post-up /sbin/sysctl -w net.bridge.bridge-nf-call-arptables=1 &> /dev/null
        post-up /sbin/sysctl -w net.bridge.bridge-nf-call-ip6tables=1  &> /dev/null
        post-up /sbin/sysctl -w net.bridge.bridge-nf-call-iptables=1  &> /dev/null
        post-up /sbin/sysctl -w net.bridge.bridge-nf-filter-pppoe-tagged=1  &> /dev/null
        post-up /sbin/sysctl -w net.bridge.bridge-nf-filter-vlan-tagged=1  &> /dev/null
        post-up /sbin/sysctl -w net.bridge.bridge-nf-pass-vlan-input-dev=1  &> /dev/null
```

Ayarlar kayıt edilir ve bridge interface UP edilir.
ifup br0

IPTables'in Bridge olarak çalışabilmesi için modüllerin aktif edilmesi gerekir.

```
if [ -f /etc/modprobe.conf ]; then
        /bin/sed -i '/^ip_tables/h;${x;/^$/{s//ip_tables/;H};x}' /etc/modprobe.conf
        /bin/sed -i '/^x_tables/h;${x;/^$/{s//x_tables/;H};x}' /etc/modprobe.conf
        /bin/sed -i '/^br_netfilter/h;${x;/^$/{s//br_netfilter/;H};x}' /etc/modprobe.conf
        /bin/sed -i '/^xt_physdev/h;${x;/^$/{s//xt_physdev/;H};x}' /etc/modprobe.conf

        /sbin/modprobe ip_tables
        /sbin/modprobe x_tables
        /sbin/modprobe br_netfilter
        /sbin/modprobe xt_physdev

else
        echo "ip_tables" >> /etc/modprobe.conf
        echo "x_tables" >> /etc/modprobe.conf
        echo "br_netfilter" >> /etc/modprobe.conf
        echo "xt_physdev" >> /etc/modprobe.conf

        /sbin/modprobe ip_tables
        /sbin/modprobe x_tables
        /sbin/modprobe br_netfilter
        /sbin/modprobe xt_physdev
fi
```

Bridge interface yapılandırması yapıldıktan sonra ve iptables modülleri aktif edildikten sonra FWBuilder ile kurallar yazılır.

1- Firewalls > Firwall_Obje > Policy > Çift Klik > Top Rule Set [ ] işareti kaldırılır.
2- Firewalls > Firwall_Obje > Sağ Klik > New Policy Rule Set denilerek yeni policy oluşturulur.
    Name: 00_SuricataNFQ
    [x] Top Rule Set işaretlenir.
3- Eklenen yeni Policy tablosuna **2** kural eklenir ve **Options** kısmında **logging off** seçilir.
4- Eklenen kuralların **Rule Options** kısmında aşağıdaki işlemler yapılır.
    a- Assume firewall is part of "any" for this rule only: **off**
    b- Stateless Rule [ ] işareti kaldırılır.
5- Eklenen kuralların **Direction** kısımları düzenlenir.
    a- 1 nolu kuralın Direction'u **OUTBOUND** olarak ayarlanır.
    b- 2 nolu kuralın Direction'u **INBOUND** olarak ayarlanır.
6- Kuralların ACTION kısmında Custom seçilir ve aşağıdaki değer yazılır.
#nproc komut ile cpu core sayısı belirlenir ve aşağıdaki kuraldaki uygun alana yazılır.
/usr/bin/nproc
Cpu Core Sayısı: 4
a- OUTBOUND ACTION
-m physdev --physdev-in eth1 --physdev-out eth2 -j NFQUEUE --queue-bypass --queue-balance 1:4
b- INBOUND ACTION
-m physdev --physdev-in eth2 --physdev-out eth1 -j NFQUEUE --queue-bypass --queue-balance 1:4

#Eğer 1 tane cpu core var ise kural'ın ACTION kısmı aşağıdaki gibi yazılır.
a- OUTBOUND ACTION
-m physdev --physdev-in eth1 --physdev-out eth2 -j NFQUEUE --queue-bypass --queue-num 1
b- INBOUND ACTION
-m physdev --physdev-in eth2 --physdev-out eth1 -j NFQUEUE --queue-bypass --queue-num 1

Not: queue-bypass parametresi ile suricata servisi çalışmıyor ise trafiği alttaki kurallardan işletmeye devam eder.

Suricata NFQUEUE Yapılandırması

Suricata servis dosyası düzenlenir ve Queue Sayısı kadar -q parametresi verilerek düzenleme yapılır.

Suricata Servis Dosyası: /lib/systemd/system/suricata.service
Bu dosyanın içerisindeki ExecStart bölümünde yer alan **--af-packet** parametresi silinir. Bunun yerine Queue sayısı kadar -q parametresi tekrarlanır.

Önceki
ExecStart=/usr/bin/suricata -D **--af-packet** -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid
Sonraki
ExecStart=/usr/bin/suricata -D **-q 1 -q 2 -q 3 -q 4** -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid
Ayarlar kayıt edildikten sonra servis yeniden başlatılır.
systemctl daemon-reload
systemctl restart suricata

2- Pulledpork Kurulum

Pulledpork kurulmadan önce suricata kurulmuş olmalıdır. Ayar dosyasındaki alanlar düzenlenir ve ansible playbook çalıştırılır.

Ayar Dosyası: /etc/ansible/roles/pulledpork/vars/main.yml

Parametreler:
pulledpork_etpro_key: open
pulledpork_rule_url: https://rules.emergingthreats.net/
pulledpork_rule_file: emerging.rules.tar.gz
pulledpork_exclude_rules:
- pass
pulledpork_update_frequency: daily

Komut: ansible-playbook /etc/ansible/playbooks/pulledpork.yml

Sonuç: Pulledpork kurulu cihazda aşağıdaki kısımlar kontrol edilir.
1- Suricata rules dizini kontrol edilir.
Komut: ls /etc/suricata/rules/emerging* | head -n 2
Sonuç: /etc/suricata/rules/emerging-activex.rules
/etc/suricata/rules/emerging-attack_response.rules

2- Periyodik imza güncelleme servisinin aktif olduğu görülür.
Komut: systemctl list-timers | grep pulledpork
Sonuç: Thu 2018-06-07 00:00:00 +03 11h left Wed 2018-06-06 00:00:05 +03 12h ago pulledpork.timer pulledpork.service

3- Snorby Kurulum

Snorby kurulmadan önce mysql server, ansible kullanılarak kurulur. Ayar dosyasındaki alanlar düzenlenir ve ansible playbook çalıştırılır.

Ayar Dosyası: /etc/ansible/roles/snorby/vars/main.yml

Parametreler:
snorby_company_name: Ahtapot
snorby_domain: ahtapot.example.com
snorby_email: snorby@example.com
snorby_password: snorby
snorby_mysql_user: snorby
snorby_mysql_password: snorby
snorby_mysql_host: localhost

Komut: ansible-playbook /etc/ansible/playbooks/snorby.yml
Sonuç: Snorby arayüzüne web üzerinden erişilir ve tanımlanan e-mail adresi ve parola ile giriş yapılır.
Snorby Web Adresi: http://cihaz_ip_adresi:3000

4- Barnyard2 Kurulum

Barnyard2 kurulmadan önce mysql, snorby ve suricata kurulmuş olmalıdır. Ayar dosyasındaki alanlar düzenlenir ve ansible playbook çalıştırılır.

Ayar Dosyası: /etc/ansible/roles/barnyard2/vars/main.yml
Parametreler:
barnyard2_interface: enp0s3
barnyard2_mysql_user: root
barnyard2_mysql_password: root
barnyard2_host: 127.0.0.1
barnyard2_sensor_name: ids_1

Komut: ansible-playbook /etc/ansible/playbooks/barnyard2.yml

Sonuç: Barnyard2 kurulduktan sonra mysql sunucuda aşağıdaki sorgu çalıştırılır. Database'de suricata imza verilerinin görüntülenmesi sistemin performansına ve imza sayısına bağlı olarak 5-15 dk arasında değişmektedir.

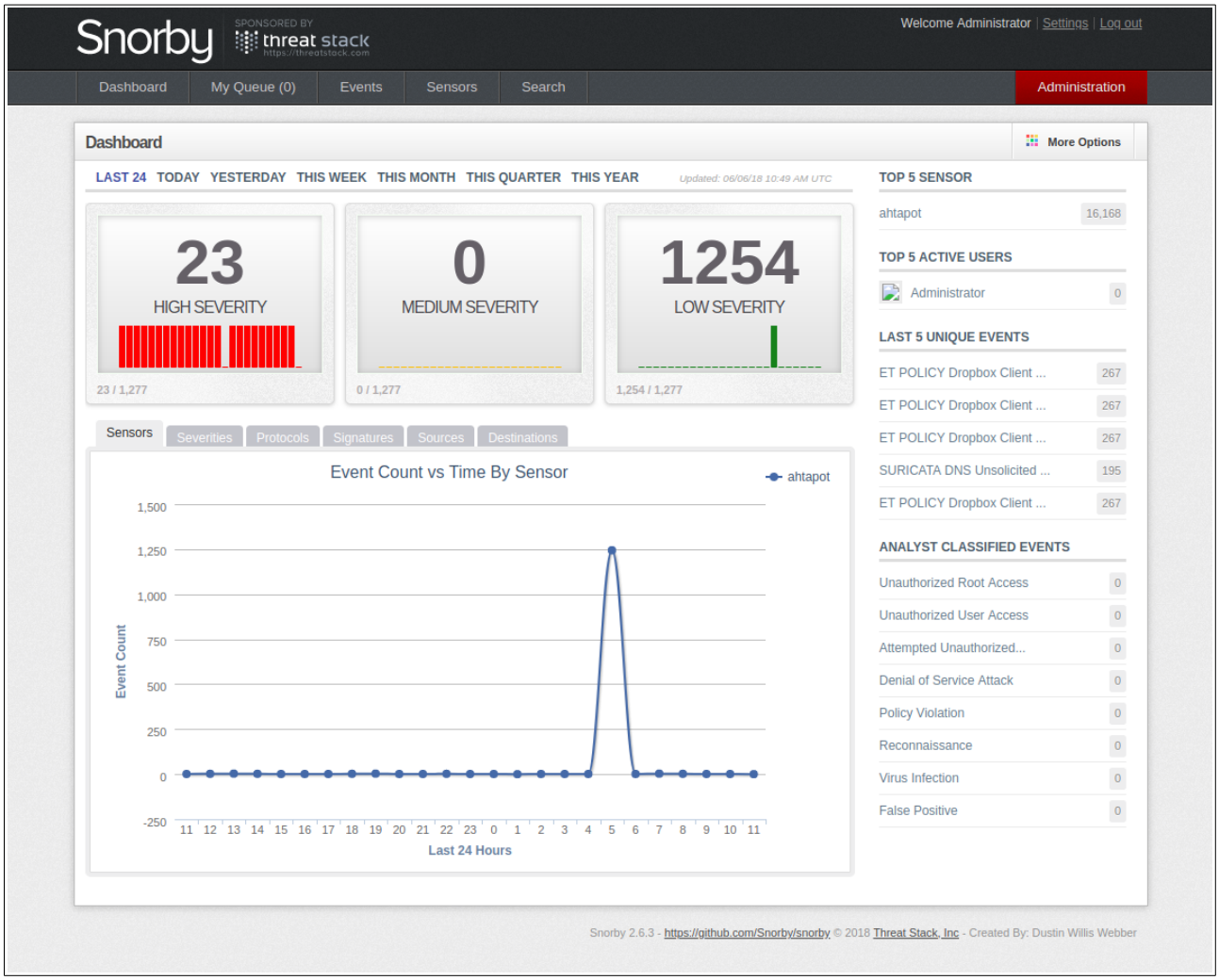
Komut: mysql -u snorby -psnorby snorby -e "select * from reference_system"
Sonuç: +---------------+-----------------+
| ref_system_id | ref_system_name |
+---------------+-----------------+
| 1 | arachNIDS |
| 2 | secunia |

5- İmza Tetikleme ve snorby üzerinde uyarı görüntüleme

Snorby, mysql, suricata, barnyard2, pulledpork kurulduktan sonra üzerinden trafik geçirilerek Snorby'de attack logları görüntülenir. Bunun için basit anlamda aşağıdaki komut bir tane pardus işletim sistem üzerinde çalıştırılır. Bu komut ile emergingthreats'in apt imzaları trigger edilir.

İmza Tetikleme: apt clean && apt update
Imza: `ET POLICY GNU/Linux APT User-Agent Outbound likely related to package managementUser-Agent Outbound likely related to package man`

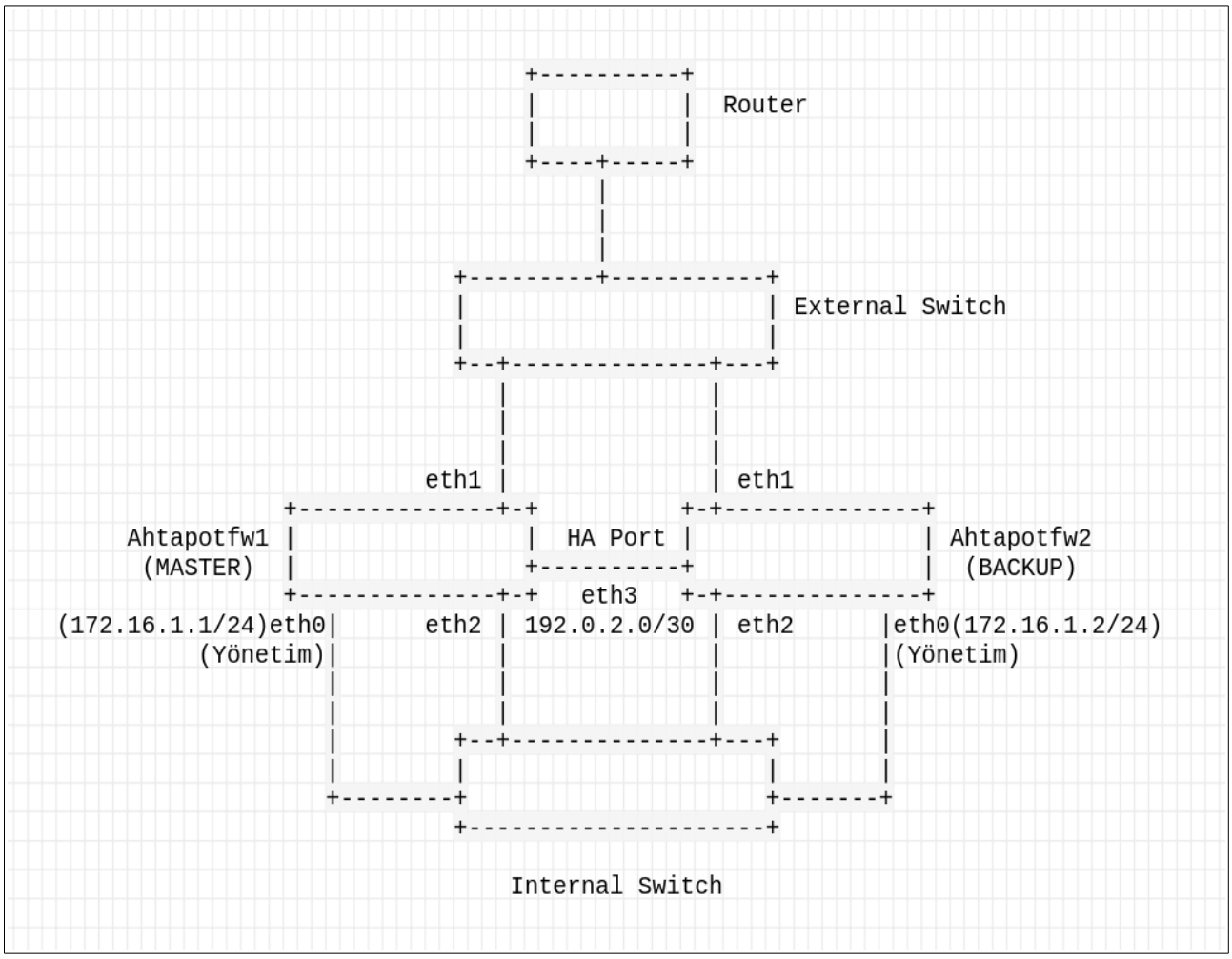
Sonuç: http://snorby_ip_adres:3000 üzerinde takip edilir.



6- Yedekli Kurulum

Yedekli Saldırı tespit ve önleme sistemi kurulumu NAT/Router ve Bridge mod olarak yapılabilir. NAT/Router mod için sistem var olan doküman takip edilir.
Bridge Mode Yedekli yapılandırma için aşağıdaki keepalived Ayar dosyası uygulanır ve Bridge inteface yapılandırması düzenlenir. Bridge mode yedekli kurulumda iki cihaz arasında Özel bir HA port kullanılmalıdır. Yönetim poru ile sistem toplamda 4 interface ile yapılandırılmalıdır.

Örnek Topoloji



MASTER cihaz keepalived yapılandırması

```
global_defs {
  router_id ahtapot1
}

vrrp_instance ahtapot {
        interface eth3 #HA interface
        state MASTER
        virtual_router_id 52
        priority 200
        authentication {
                auth_type PASS
                auth_pass ahtapot
        }

        #Bridge üyesi fiziksel interface'ler
        track_interface {
                eth1
                eth2
        }
}
```

MASTER cihaz Bridge interface yapılandırması.

Bridge Mod IPS bölümündeki bridge interface yapılandırması uygulanır. Bu yapılandırmaya ek olarak aşağıdaki parametre eklenir.

Bu ayar sadece master cihazda yapılır. Backup olan cihazda böyle bir yapılandırma **yapılmaz**.

```
        #High Availability config
        bridge_bridgeprio 30000 # Only Master Node
```

MASTER HA Port Ayarları

auto eth3

iface eth3 inet static

address 192.0.2.1

netmask 255.255.255.252

BACKUP cihaz keepalived yapılandırması

```
global_defs {
  router_id ahtapot2
}

vrrp_instance ahtapot {
        interface eth3 #HA interface
        state BACKUP
        virtual_router_id 52
        priority 100
        authentication {
                auth_type PASS
                auth_pass ahtapot
        }

        #Bridge üyesi fiziksel interface'ler
        track_interface {
                eth1
                eth2
        }
}
```

BACKUP HA Port Ayarları

auto eth3

iface eth3 inet static

address 192.0.2.2

netmask 255.255.255.252

HA portlar her iki cihazda da UP edilir.

ifup eth3

Diğer kısımlar NAT/Router mode yapılandırma ve test prosedürü ile aynıdır. Link