



PANDUAN PRAKTIS DAN LENGKAP MEMBUAT GATEWAY INTERNET LINUX ARINET

Table of Contents

Kata Pengantar.....	3
Sekilas mengenai Fajar Priyanto.....	4
Lisensi.....	5
Feature-feature utama internet gateway Arinet.....	6
Shorewall (www.shorewall.net)	6
Squid (www.squid-cache.org)	6
SquidGuard (www.squidguard.org)	6
Sarg (sarg.sourceforge.net).....	6
Denyhosts (denyhosts.sourceforge.net)	7
Cacti (www.cacti.net).....	7
Asumsi dan persyaratan:.....	7
Mulai instalasi: Centos 4.4.....	8
Instalasi dan Konfigurasi Shorewall	9
Konfigurasi Squid.....	14
Instalasi dan Konfigurasi SquidGuard	17
Instalasi dan Konfigurasi SARG	21
Instalasi dan Konfigurasi Cacti.....	24
Instalasi dan Konfigurasi DenyHosts	28
Security Considerations.....	31
Kesimpulan dan Penutup.....	33
Changelog.....	34

Kata Pengantar

Kehandalan Linux sebagai operating system menjadikannya banyak digunakan sebagai server, seperti DNS server, Web server, Mail server, dll. Demikian pula sebagai gateway internet, banyak perusahaan yang walaupun masih banyak server-servernya menggunakan produk proprietary namun untuk urusan gateway mereka mempercayakannya kepada Linux.

Belakangan ini Linux semakin mudah digunakan, dimana untuk menjadikan Linux sebagai gateway internet sudah sangat mudah. Beberapa distro seperti OpenSuse, Ubuntu, dan Mandriva telah memiliki menu di control center-nya untuk mensetup gateway. Namun memang tetap harus diakui bahwa untuk menjadikannya sebuah gateway yang cukup lengkap dan secure masih relatif “gampang-gampang susah”.

Oleh karena itu penulis berusaha menyusun panduan ini. Pertanyaannya sekarang adalah “Mengapa membuat panduan ini?” Dan “Mengapa pula menyediakannya secara free?”

Jawabannya adalah sangat pribadi. Sudah sejak lama penulis melihat di forum-forum internet, dan mailing list Linux ada yang bertanya bagaimana caranya membuat gateway internet. Oleh karena itu, penulis terpanggil untuk membantu komunitas dengan jalan menuliskan panduan ini. Diharapkan dengan adanya panduan ini, maka daripada menghabiskan uang sampai jutaan rupiah untuk membeli produk proprietary, diharapkan kita semua dapat “mulai” membuat sendiri gateway internet menggunakan Linux. Mengapa menggunakan kata “mulai”? Sebab penulis mengakui panduan ini tidaklah sempurna. Masih banyak yang harus ditingkatkan. Oleh karena itu, sangat diharapkan kelak para pembaca dapat memberikan masukan-masukan kepada penulis demi perbaikan panduan ini. Silahkan sampaikan saran/kritik di Forum Arinet (<http://linux2.arinet.org>).

Terima kasih kepada banyak pihak yang telah memberikan inspirasi dan dukungannya kepada penulis. Pak Bambang Gunawan, Oom Onno W. Purbo, I Made Wiryana, Pak Rusmanto, komunitas AsiaSource, Linux.or.id, dan rekan-rekan kerja di Bajau. Terima kasih pula kepada Devi istri tercinta atas pengertian dan supportnya selama ini.

Selamat mencoba.

Ilmu Pengetahuan adalah Milik Bersama.

Merdeka!

Fajar Priyanto

fajarpri@arinet.org

ym: fajarpri

Bukit Sentul, 29 April 2007

Indonesia

Sekilas mengenai Fajar Priyanto



Ia memulai petualangan Linux dan OpenSourceny di tahun 2001 ketika ia baru saja mendapatkan sertifikasi MCP (Microsoft Certified Professional) di Windows 2000 Server. Setelah itu, ia menjadi semakin tertarik kepada Linux. Ia menjabat sebagai Kepala Bagian Sistem dan Teknologi BPK Penabur dari awal 2002 sampai dengan akhir 2005. Pada saat itu ia berkesempatan mensetup beberapa server Linux sekaligus sebagai SysAdmin collocated servernya. Selanjutnya ia bergabung dengan Astrido Group sebagai IT Section Head dimana ia memimpin sebuah team IT untuk memberikan support kepada belasan cabang. Ia mensetup pula beberapa server Linux mulai dari internet gateway, dns, web, ftp, mail, file, fax, dan database server.

Selanjutnya ia mendapatkan sertifikasi RedHat Certified Engineer di pertengahan tahun 2006 dan sekarang bekerja sebagai salah seorang engineer dan instruktur di Bajau. Ia sangat menyukai mengajar dan menulis. Di waktu luangnya, ia menulis artikel dan tutorial Linux di website pribadinya <http://linux2.arinet.org>. Kini telah terdapat sekitar 4500 registered users dengan rata-rata pertumbuhan belasan user setiap harinya.



Pengalamannya meliputi installing, setting up dan maintaining server Linux dengan service-service: firewall, proxy, dns, web, ftp, database, fax, clustering, dan lain-lain. Baru-baru ini ia dikirim ke RedHat Training Center di Singapore untuk mengikuti kelas RedHat Enterprise Storage Management yang mengupas penggunaan RedHat Cluster Suite dan Global File System. Sertifikasinya dapat dilihat di <https://www.redhat.com/training/certification/> dan masukkan nomor RHCEnya 804006611022453.

Ia selalu percaya kepada semangat dan nilai-nilai OpenSource. Di tahun 2004 ia menjadi salah satu peserta dari Indonesia untuk menghadiri AsiaSource di Bangalore, India. Kegiatan selama 10 hari ini merupakan gathering para aktivis OpenSource dari seluruh Asia yang disponsori oleh UNDP dan organisasi OpenSource lainnya. Pepatah favoritnya mengenai OpenSource adalah “Balaslah jasa para senior yang telah membantumu dengan jalan membantu para newbie sesudah kamu” (dari I Made Wiryana).

Lisensi

Panduan ini dirilis menggunakan Lisensi Creative Common License
(<http://creativecommons.org/licenses/by-nc-sa/3.0/>)

Silahkan di klik link di atas untuk melihat secara detail, tapi secara singkat memiliki arti:

Kamu bebas untuk:

1. Mengcopy, mendistribusikan, dan mempraktekkannya
2. Melakukan perubahan atas aslinya.

Namun dengan syarat:

1. Kamu harus menyebutkan penulis aslinya.
2. Tidak digunakan untuk keperluan komersial.
3. Jika kamu melakukan perubahan apapun, atau melakukan pekerjaan atas dasar panduan ini, kamu harus JUGA menggunakan lisensi yang IDENTIK dengan lisensi disini.



To Share - to copy, distribute and transmit the work



To Remix - to adapt the work



Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial. You may not use this work for commercial purposes.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Feature-feature utama internet gateway Arinet

1. Firewall dengan Shorewall
2. Transparan Proxy dengan Squid
3. Web filtering dengan SquidGuard
4. Proxy reporting dengan Sarg
5. Automatic security blocking dengan Denyhosts
6. Web based monitoring dengan Cacti

Shorewall (www.shorewall.net)



Di Linux kita mengenal iptables sebagai salah satu modul dari kernel untuk mengatur koneksi TCP/IP. Terdapat banyak tutorial mengenai iptables di internet, namun diperlukan pengetahuan yang cukup mendalam mengenai iptables beserta seluruh syntax dan kemampuannya untuk bisa menghasilkan policy dan rules yang secure dan cocok dengan yang kita inginkan. Oleh karena itu digunakanlah Shorewall ini. Shorewall membantu kita di dalam mengatur iptables dengan format file konfigurasi yang mudah dipahami. Melalui Shorewall, kita dapat dengan mudah mengatur port-port apa saja yang ingin kita buka, siapa yang boleh mengaksesnya, permission berdasarkan direction trafik tersebut, port forwarding, dll.

Squid (www.squid-cache.org)



Squid boleh dibilang merupakan salah satu server proxying yang paling populer di dunia Linux. Selain merupakan produk yang stable, ia memiliki banyak feature seperti: full access control list, hierarchy caching, dns caching, snmp support, dll.

SquidGuard (www.squidguard.org)



SquidGuard dapat dianalogikan seperti fungsi Shorewall terhadap iptables. SquidGuard merupakan frontend dari Squid, ditambah dengan database kategori-kategori website seperti website porno, ads, hacking, dll. Jadi kita dapat membuat ACL (access control list) untuk otomatis memblokir website-website yang masuk kategori porno, dll.

Sarg (sarg.sourceforge.net)



Sarg atau Squid Access Report Generator adalah script yang bisa menganalisa access.log dari Squid dan kemudian membuatnya dalam tampilan web. Melalui report ini kita bisa mengetahui penggunaan web yang melalui proxy kita berdasarkan user, jam, besarnya data, dll.

Denyhosts (denyhosts.sourceforge.net)

DenyHOSTS Denyhosts merupakan script yang menganalisa log file Linux kita untuk melihat apakah ada yang mencoba login ke server kita namun gagal. Berdasarkan dari analisa ini kita dapat melakukan blocking terhadap IP dari si hacker sehingga ia tidak akan dapat melakukan login kembali. Di zaman internet yang penuh resiko sekarang ini, alangkah baiknya jika kita menerapkan script automatic blocking ini.

Cacti (www.cacti.net)



Cacti adalah script untuk memonitor berbagai macam hal yang ada di dalam gateway kita, seperti: traffic monitoring, CPU load, memory usage, dll. Sangat berguna untuk mengetahui beban kerja dan performance Linux kita.

Asumsi dan persyaratan:

Walaupun panduan ini dibuat sedemikian rupa dan step-by-step agar mudah diikuti, namun diharapkan kamu harus setidaknya telah familiar dengan command-command Linux seperti mengedit file konfigurasi, mengetahui konsep jaringan, biasa menginstall Linux, menjalankan dan mematikan service, dll.

PC gateway kita sebaiknya memiliki spesifikasi hardware minimum sebagai berikut:

- RAM 128MB (semakin besar semakin baik)
- HDD 6GB (bila ada gunakan 2 buah untuk dibuat mirror)
- 2 buah network card yang di support Linux
- CD ROM untuk installasi

Mulai instalasi: Centos 4.4

Dari sekian banyak distro Linux yang ada, mengapa menggunakan Centos 4.4? Sebenarnya tidak masalah jika menggunakan distro-distro lainnya, seperti OpenSuse, Ubuntu, Mandriva, Slackware, Debian, dll.

Penyebab utama penulis menggunakan Centos 4.4 adalah:

1. Free. Ia dapat didownload dari www.centos.org
2. Ia merupakan kompilasi ulang oleh komunitas dari sebuah distro Linux yang sangat terkenal. Didukung oleh team developer dan komunitas yang sangat aktif dan kompeten di bidangnya.
3. Merupakan salah satu distro Linux yang banyak dipakai di dunia, baik oleh komunitas maupun lingkungan corporate.
4. Paket-paket RPM OpenSourcenya banyak tersedia di internet.

Install Centos secara biasa, pastikan paket-paket Xwindow, Samba, Printing, dan paket-paket desktop lainnya **tidak** terinstall demi alasan keamanan. Install-lah paket library development untuk kemudahan.

Sebagai bahan acuan dapat dilihat artikel menginstall Centos lengkap dengan LVM dan RAID-1 di

http://linux2.arinet.org/index.php?option=com_content&task=view&id=123&Itemid=36

Untuk komponen-komponen servernya, install:

1. bind-caching-nameserver
2. squid
3. apache beserta modul-modulnya: php-devel, php-mysql, php-snmp, php-gd
4. mysql, mysql-server
5. snmp, net-snmp-utils

Sebagai gateway, PC Linux kita mempunyai 2 buah network card. Dimana yang satu (eth0) terkoneksi ke Internet (misalnya 202.137.123.240) dan yang satu lagi (eth1) terkoneksi ke LAN internal kita (misalnya 10.0.0.250).

Bila telah selesai menginstall Centos, maka kita akan dapat mulai menginstall dan mensetup gateway.

Instalasi dan Konfigurasi Shorewall

Download source RPM shorewall dari www.shorewall.net.

```
# wget http://www.invocha.ch/pub/packages/shorewall/3.4/shorewall-3.4.1/shorewall-3.4.1-3.src.rpm
```

Compile shorewall:

```
# rpmbuild --rebuild shorewall-3.4.1-3.src.rpm
Installing shorewall-3.4.1-3.src.rpm
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.76342
+ umask 022
+ cd /usr/src/redhat/BUILD
+ LANG=C
+ export LANG
+ unset DISPLAY
+ cd /usr/src/redhat/BUILD

.....
.....
```

```
Requires: /bin/sh /etc/redhat-release config(shorewall-lite) = 3.4.1-3 iproute
iptables
Checking for unpackaged file(s): /usr/lib/rpm/check-files /var/tmp/shorewall-3.4.1-
root
Wrote: /usr/src/redhat/RPMS/noarch/shorewall-3.4.1-3.noarch.rpm
Wrote: /usr/src/redhat/RPMS/noarch/shorewall-lite-3.4.1-3.noarch.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.65085
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd shorewall-3.4.1
+ '[' /var/tmp/shorewall-3.4.1-root '!=' / ']'
+ /bin/rm -rf /var/tmp/shorewall-3.4.1-root
+ exit 0
Executing(--clean): /bin/sh -e /var/tmp/rpm-tmp.65085
+ umask 022
+ cd /usr/src/redhat/BUILD
+ rm -rf shorewall-3.4.1
+ exit 0
```

Install shorewall:

```
# rpm -ivh /usr/src/redhat/RPMS/noarch/shorewall-3.4.1-3.noarch.rpm
Preparing... ##### [100%]
 1:shorewall ##### [100%]
```

Setting Shorewall:

Edit file-file ini di /etc/shorewall

a. shorewall.conf

```
-----  
STARTUP_ENABLED=yes  
IP_FORWARDING=On
```

b. interfaces

```
-----  
#ZONE    INTERFACE    BROADCAST    OPTIONS  
net      eth0          detect  
loc      eth1          detect
```

c. zones

```
-----  
#ZONE    TYPE          OPTIONS      IN           OUT  
#                   OPTIONS      OPTIONS  
fw       firewall  
net      ipv4  
loc      ipv4
```

d. masq

```
-----  
#INTERFACE    SOURCE    ADDRESS    PROTO    PORT(S)    IPSEC  
eth0          eth1
```

e. policy

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

f. rules

```
-----  
# Transparent proxy  
REDIRECT    loc          3128        tcp        www        -  
ACCEPT      $FW          net         tcp        www  
  
# Accept DNS connections from the firewall to the network  
DNS/ACCEPT  $FW          net  
DNS/ACCEPT  net          $FW  
DNS/ACCEPT  loc          net  
DNS/ACCEPT  $FW          loc  
DNS/ACCEPT  loc          $FW  
  
# Accept SSH connections from the local network for administration  
SSH/ACCEPT  loc          $FW
```

```
SSH/ACCEPT      net          $FW
SSH/ACCEPT      loc          net
SSH/ACCEPT      $FW          loc

# Allow Ping from the local network
Ping/ACCEPT      loc          $FW

# Reject Ping from the "bad" net zone.. and prevent your log from being flooded..
Ping/REJECT      net          $FW
ACCEPT           $FW          loc          icmp
ACCEPT           $FW          net          icmp
ACCEPT           loc          $FW          icmp

# Web
Web/ACCEPT       loc          net

# Mail
POP3/ACCEPT      loc          net
SMTP/ACCEPT      loc          net
ICQ/ACCEPT       loc          net

# Others rules
ACCEPT           loc          net          tcp          2082,2095
ACCEPT           loc          net          tcp          5050
```

Testing Shorewall.

Setelah terinstall, kita dapat mulai melihat apakah shorewall kita sudah dapat berjalan. Buka 2 buah terminal. Di satu terminal kita lakukan: `tail -f /var/log/messages` dan di terminal lain kita ketik:

```
# chkconfig shorewall on
# service shorewall start
```

Perhatikan di terminal yang satunya akan terlihat seperti ini:

```
Apr  8 01:06:00 gateway shorewall: Compiling...
Apr  8 01:06:01 gateway shorewall: Initializing...
Apr  8 01:06:02 gateway shorewall: Determining Zones...
Apr  8 01:06:02 gateway shorewall:   IPv4 Zones: net loc
Apr  8 01:06:02 gateway shorewall:   Firewall Zone: fw
Apr  8 01:06:02 gateway shorewall: Validating interfaces file...
Apr  8 01:06:02 gateway shorewall: Validating hosts file...
Apr  8 01:06:02 gateway shorewall: Pre-processing Actions...
Apr  8 01:06:02 gateway shorewall:   Pre-processing
/usr/share/shorewall/action.Drop...
Apr  8 01:06:02 gateway shorewall:   Pre-processing
/usr/share/shorewall/action.Reject...
Apr  8 01:06:02 gateway shorewall: Validating Policy file...
Apr  8 01:06:02 gateway shorewall: Determining Hosts in Zones...
Apr  8 01:06:02 gateway shorewall:   net Zone: eth0:0.0.0.0/0
```

```
Apr  8 01:06:02 gateway shorewall:    loc Zone: eth1:0.0.0.0/0
Apr  8 01:06:02 gateway shorewall: Deleting user chains...
Apr  8 01:06:02 gateway shorewall: Compiling /etc/shorewall/routestopped ...
Apr  8 01:06:02 gateway shorewall: Creating Interface Chains...
Apr  8 01:06:02 gateway shorewall: Compiling Common Rules
Apr  8 01:06:02 gateway shorewall: Compiling IP Forwarding...
Apr  8 01:06:03 gateway shorewall: Compiling /etc/shorewall/rules...
Apr  8 01:06:04 gateway shorewall: Compiling Actions...
Apr  8 01:06:04 gateway shorewall: Compiling /usr/share/shorewall/action.Drop for
Chain Drop...
Apr  8 01:06:04 gateway shorewall: Compiling /usr/share/shorewall/action.Reject for
Chain Reject...
Apr  8 01:06:05 gateway shorewall: Compiling /etc/shorewall/policy...
Apr  8 01:06:05 gateway shorewall: Compiling Masquerading/SNAT
Apr  8 01:06:05 gateway shorewall: Compiling Traffic Control Rules...
Apr  8 01:06:05 gateway shorewall: Compiling Rule Activation...
Apr  8 01:06:06 gateway shorewall: Shorewall configuration compiled to
/var/lib/shorewall/.restart
Apr  8 01:06:06 gateway shorewall: Processing /etc/shorewall/params ...
Apr  8 01:06:06 gateway shorewall: Restarting Shorewall....
Apr  8 01:06:06 gateway shorewall: Initializing...
Apr  8 01:06:07 gateway shorewall: Processing /etc/shorewall/init ...
Apr  8 01:06:07 gateway shorewall: Clearing Traffic Control/QOS
Apr  8 01:06:07 gateway shorewall: Deleting user chains...
Apr  8 01:06:07 gateway shorewall: Processing /etc/shorewall/continue ...
Apr  8 01:06:07 gateway shorewall:    WARNING: DISABLE_IPV6=Yes in shorewall.conf
but this system does not appear to have ip6tables
Apr  8 01:06:07 gateway shorewall: Enabling Loopback and DNS Lookups
Apr  8 01:06:07 gateway shorewall: Creating Interface Chains...
Apr  8 01:06:07 gateway shorewall: Setting up SMURF control...
Apr  8 01:06:07 gateway shorewall: Processing /etc/shorewall/initdone ...
Apr  8 01:06:07 gateway shorewall: Setting up Black List...
Apr  8 01:06:07 gateway shorewall: Setting up ARP filtering...
Apr  8 01:06:07 gateway shorewall: Setting up Accept Source Routing...
Apr  8 01:06:07 gateway shorewall: IP Forwarding Enabled
Apr  8 01:06:07 gateway shorewall: Setting up SYN Flood Protection...
Apr  8 01:06:07 gateway shorewall: Setting up Rules...
Apr  8 01:06:08 gateway shorewall: Setting up Actions...
Apr  8 01:06:08 gateway shorewall: Creating action chain Drop
Apr  8 01:06:08 gateway shorewall: Creating action chain Reject
Apr  8 01:06:08 gateway shorewall: Creating action chain dropBcast
Apr  8 01:06:08 gateway shorewall: Creating action chain dropInvalid
Apr  8 01:06:08 gateway shorewall: Creating action chain dropNotSyn
Apr  8 01:06:08 gateway shorewall: Applying Policies...
Apr  8 01:06:08 gateway shorewall: Setting up Masquerading/SNAT...
Apr  8 01:06:08 gateway root: Shorewall restarted
Apr  8 01:06:08 gateway shorewall: Activating Rules...
Apr  8 01:06:08 gateway shorewall: Processing /etc/shorewall/start ...
Apr  8 01:06:08 gateway shorewall: Processing /etc/shorewall/started ...
Apr  8 01:06:08 gateway shorewall: done.
Apr  8 01:06:09 gateway shorewall: shorewall startup succeeded
```

Kemudian dari PC lain di network yang terhubung dengan gateway kita, kita testing dengan melakukan port scanning:

```
# nmap 10.0.0.250 (ip gateway kita contohnya)
```

Di terminal yang tadi akan terlihat:

```
Apr  8 01:04:55 gateway kernel: Shorewall:loc2net:DROP:IN=eth1 OUT=eth0  
SRC=10.0.0.101 DST=64.202.165.92 LEN=60 TOS=0x00 PREC=0x00 TTL=63 ID=46867 DF  
PROTO=TCP SPT=48398 DPT=995 WINDOW=5840 RES=0x00 SYN URGP=0  
Apr  8 01:04:55 gateway kernel: Shorewall:loc2net:DROP:IN=eth1 OUT=eth0  
SRC=10.0.0.101 DST=64.202.165.92 LEN=60 TOS=0x00 PREC=0x00 TTL=63 ID=27479 DF  
PROTO=TCP SPT=48399 DPT=995 WINDOW=5840 RES=0x00 SYN URGP=0
```

Konfigurasi Squid.

Kita memanfaatkan squid yang bawaan dari Centos saja yang telah kita install di awal.

Konfigurasi squid ada di `/etc/squid/squid.conf`

Edit file tersebut, cari baris yang ada kata-kata: `our_networks`

Dan tambahkan:

```
our_networks 10.0.0.0/24
http_access allow our_networks
```

10.0.0.0/24 ini adalah network yang ingin kita perbolehkan mengakses internet melalui squid. Sesuaikan angkanya dengan kondisi network kamu.

Kemudian di baris paling bawah dari `squid.conf`, kita tambahkan options ini:

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

Option-option di atas tersebut berguna untuk mengaktifkan fungsi transparent proxy gateway kita.

Aktifkan squid agar hidup secara default ketika menjalankan Linux:

```
# chkconfig squid on
```

=====
Setting DNS

Sebelum bisa melakukan browsing, kita harus menyetel DNS dahulu. DNS ini adalah optional, dalam arti kamu dapat menggunakan DNS ISP kamu saja atau menyetelnya sendiri. Bila ingin menggunakan DNS ISP kamu, isikan di `/etc/resolv.conf` :

```
nameserver 202.158.3.7 (misalnya)
```

Atau bila ingin menggunakan DNS sendiri, edit file `/etc/named.conf`, dan aktifkan/tambahkan ini di bagian options:

```
forwarders { 202.152.3.7; };
```

Dan di `/etc/resolv.conf`:

```
nameserver 127.0.0.1
```

Setel agar DNS berjalan secara otomatis pada startup:


```
# chkconfig named on
# service named restart
```

Cek DNS:

```
# dig www.linux.or.id
```

Hasilnya harus terlihat seperti ini:

```
; <<>> DiG 9.2.4 <<>> www.linux.or.id
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51340
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 6, ADDITIONAL: 6

;; QUESTION SECTION:
www.linux.or.id.                IN      A

;; ANSWER SECTION:
www.linux.or.id.                82078   IN      A      67.19.121.27

;; AUTHORITY SECTION:
linux.or.id.                    12327   IN      NS      dns3.client.org.
linux.or.id.                    12327   IN      NS      dns4.client.org.
linux.or.id.                    12327   IN      NS      dns5.client.org.
linux.or.id.                    12327   IN      NS      dns6.client.org.
linux.or.id.                    12327   IN      NS      dns1.client.org.
linux.or.id.                    12327   IN      NS      dns2.client.org.

;; ADDITIONAL SECTION:
dns1.client.org.                10209   IN      A      67.19.104.99
dns2.client.org.                10209   IN      A      67.19.116.203
dns3.client.org.                10225   IN      A      67.19.121.11
dns4.client.org.                10225   IN      A      67.19.121.27
dns5.client.org.                16782   IN      A      67.19.121.35
dns6.client.org.                15456   IN      A      67.19.121.59

;; Query time: 81 msec
;; SERVER: 192.168.0.254#53(192.168.0.254)
;; WHEN: Tue May 1 15:03:16 2007
;; MSG SIZE rcvd: 269
```

Ok, berarti DNS sudah dapat berjalan dengan baik.

Testing squid:

```
# chkconfig squid on
# service squid restart
```

Setup sebuah PC client di network agar menggunakan IP gateway kita sebagai gateway dia, dan kalau

perlu gunakan pula IP gateway kita sebagai DNS server dia.

Kemudian hidupkan browser, dan cobalah browsing internet. Di gateway, kita monitor melalui terminal:

```
# tail -f /var/log/squid/access.log
```

Mesti terlihat aktivitas terjadi di file log tersebut:

```
1175971357.637 3624 10.0.0.101 TCP_MISS/200 4116 GET
http://www.youtube.com/set_awesome? - DIRECT/208.65.153.253 text/xml
1175971388.078 251945 10.0.0.101 TCP_MISS/200 7674038 GET http://sjc-
v46.sjc.youtube.com/get_video? - DIRECT/64.15.124.214 video/flv
1175971400.358 1144 10.0.0.101 TCP_MISS/200 2694 GET http://sjl-
static2.sjl.youtube.com/vi/9U1-m9-pXQk/2.jpg - DIRECT/208.65.153.10 image/jpeg
```

Bila ternyata dari client belum bisa melakukan browsing dan di log file squid tidak terdapat aktivitas apapun, berarti ada kesalahan setup di depan. Cek ulanglah lagi.

Bila dari client sudah dapat melakukan browsing, artinya Selamat! Linux kita sebenarnya telah dapat menjadi sebuah gateway dan proxy. Namun mari kita lengkapi lagi dia dengan feature-feature cool lainnya.

Instalasi dan Konfigurasi SquidGuard

Sebenarnya tersedia squidGuard dalam bentuk RPM, namun ada baiknya kita gunakan yang dari source tar.gz sebab versinya lebih baru dan mengandung patch-patch untuk feature tambahan seperti mengatur ACL user berdasarkan banyaknya jam, dll.

Download squidGuard:

```
# wget ftp://ftp.univ-tlse1.fr/pub/contrib\_ut1/squidguard/squidGuard-1.2.10.tar.gz
```

Extract:

```
# tar zxvf squidGuard-1.2.10.tar.gz
```

Compile dan install:

```
# mkdir /var/log/squidguard  
# touch /var/log/squidguard/squidGuard.log  
# cd squidGuard-1.2.10  
# ./configure --with-sg-config=/etc/squid/squidguard.conf --with-sg-logdir=/var/log/squidguard -with-sg-dbhome=/var/lib/squidguard/db
```

```
# make  
# make test  
# make install
```

```
# chown squid.squid /var/log/squidguard/squidGuard.log
```

Pastikan bahwa di akhir baris dari file `/etc/squid/squid.conf` terdapat baris ini, bila tidak ada buatlah:

```
redirect_program /usr/local/bin/squidGuard -c /etc/squid/squidguard.conf
```

Download database squidGuard (save misalnya di direktori `/root`):

```
# wget ftp://ftp.univ-tlse1.fr/pub/reseau/cache/squidguard_contrib/blacklists.tar.gz
```

```
# mkdir /var/lib/squidguard/db  
# cd /var/lib/squidguard/db  
# tar zxvf /root/blacklists.tar.gz
```

Maka akan tercreate direktori blacklists di dalam `/var/lib/squidguard/db`

```
# chown -R squid.squid /var/lib/squidguard
```

Create/edit file `/etc/squid/squidguard.conf` dengan isi sebagai berikut:

```
#
```

```
# CONFIG FILE FOR SQUIDGUARD
#

dbhome /var/lib/squidguard/db/blacklists
logdir /var/log/squidguard

#
# TIME RULES:
# abbrev for weekdays:

# s = sun, m = mon, t =tue, w = wed, h = thu, f = fri, a = sat

time leisure-time {
    weekly mtwhfa 00:01 - 08:30 12:00 - 13:00 17:30 - 24:00
}

# -----
# SOURCE ADDRESSES:
# -----

# user network kita.
src our_network {
    ip 10.0.0.101 #fajar
}
#
# DESTINATION CLASSES:
#

dest adult {
    domainlist adult/domains
    urllist adult/urls
    expressionlist adult/expressions
}

dest audio-video {
    domainlist audio-video/domains
    urllist audio-video/urls
}

dest hacking {
    domainlist hacking/domains
    urllist hacking/urls
}

dest warez {
    domainlist warez/domains
    urllist warez/urls
}

dest ads {
```

```
domainlist ads/domains
urllist ads/urls
}

dest aggressive {
    domainlist aggressive/domains
    urllist aggressive/urls
}

dest drugs {
    domainlist drugs/domains
    urllist drugs/urls
}

dest gambling {
    domainlist gambling/domains
    urllist gambling/urls
}

acl {

    our_network {
        pass !adult any
        redirect http://localhost/maaf.html
    }

    default {
        pass none
        redirect http://localhost/maaf-unknown.html
    }
}
```

Buatlah 2 buah file html biasa:

1. maaf.html dengan isi: Maaf Akses Website ini diblock. Dan taruh file html itu di /var/www/html/
2. maaf-unknown.html dengan isi: Maaf, hanya user yang terdaftar yang dapat browsing internet, dan taruh di /var/www/html/
3. Oya, pastikan apache dapat berjalan otomatis:
4. # chkconfig httpd on
5. Edit file /etc/httpd/conf.d/welcome.conf dan beri tanda pagar di dalam barisnya.
6. Jalankan apache:
7. # service httpd restart

Selanjutnya kita build database squidGuard tersebut sambil memonitor log filenya agar kita tahu bila ada error. Buka 2 terminal, di terminal 1 kita jalankan command:

```
# tail -f /var/log/squidguard/squidGuard.log
```

Dan di terminal satunya lagi:

```
# squidGuard -C all
```

Di /var/log/squidguard/squidGuard.log itu akan dapat kita lihat:

```
2007-04-08 14:34:02 [15197] New setting: dbhome: /var/lib/squidguard/db/blacklists
2007-04-08 14:34:02 [15197] New setting: logdir: /var/log/squidguard
2007-04-08 14:34:02 [15197] init domainlist
/var/lib/squidguard/db/blacklists/adult/domains
2007-04-08 14:34:39 [15197] create new dbfile
/var/lib/squidguard/db/blacklists/adult/domains.db
2007-04-08 14:34:52 [15197] init urllist
/var/lib/squidguard/db/blacklists/gambling/urls
2007-04-08 14:34:52 [15197] create new dbfile
/var/lib/squidguard/db/blacklists/gambling/urls.db
2007-04-08 14:41:45 [15204] squidGuard 1.2.10 started (1176018056.576)
2007-04-08 14:41:45 [15204] db update done
2007-04-08 14:41:45 [15204] squidGuard stopped (1176018105.304)
```

Jangan lupa setiap habis melakukan squidGuard -C, kita jalankan command ini:

```
# chown -R squid.squid /var/lib/squidguard
```

Testing SquidGuard. Cobalah dari PC client mencoba membuka website, misalnya: www.playboy.com.
Maka akan muncul ACCESS DENIED.

Instalasi dan Konfigurasi SARG

Download Sarg:

```
# wget http://optusnet.dl.sourceforge.net/sourceforge/sarg/sarg-2.2.3.1.tar.gz
```

Extract dan configure:

```
# tar zxvf sarg-2.2.3.1.tar.gz
# cd sarg-2.2.3.1
# ./configure
# make
# make install

creating /usr/local/man/man1
creating /usr/local/sarg
Creating /usr/local/sarg/languages
Creating /usr/local/sarg/fonts
cp sarg /usr/bin/sarg
chmod 755 /usr/bin/sarg
cp sarg.1 /usr/local/man/man1/sarg.1
chmod 755 /usr/local/man/man1/sarg.1
cp /usr/local/sarg/sarg.conf
cp -r ./languages /usr/local/sarg;
cp ./exclude_codes /usr/local/sarg;
cp ./user_limit_block /usr/local/sarg;
cp -r ./images /usr/local/sarg;
cp -r ./sarg-php /usr/local/sarg;
cp -r ./fonts /usr/local/sarg;
cp -r ./css.tpl /usr/local/sarg;
```

Siapkan script agar sarg dapat menghasilkan reportnya secara harian, mingguan, dan bulanan.

```
# vi /etc/cron.daily/sarg
```

Dengan isi:

```
#!/bin/bash

# Get yesterday's date
YESTERDAY=$(date --date "1 days ago" +%d/%m/%Y)

exec /usr/bin/sarg \
    -f /etc/sarg/sarg.conf \
    -o /var/www/html/sarg/daily \
    -d $YESTERDAY

exit 0
```

```
# vi /etc/cron.weekly/sarg
```

Dengan isi:

```
#!/bin/bash
LOG_FILES=
if [ -s /var/log/squid/access.log.1.gz ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log.1.gz"
fi
if [ -s /var/log/squid/access.log ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log"
fi

# Get yesterday's date
YESTERDAY=$(date --date "1 days ago" +%d/%m/%Y)

# Get one week ago date
WEEKAGO=$(date --date "7 days ago" +%d/%m/%Y)

exec /usr/bin/sarg \
    $LOG_FILES \
    -f /etc/sarg/sarg.conf \
    -o /var/www/html/sarg/weekly \
    -d $WEEKAGO-$YESTERDAY &>/dev/null
exit 0
```

```
# vi /etc/cron.monthly/sarg
```

Dengan isi:

```
#!/bin/bash
LOG_FILES=
if [ -s /var/log/squid/access.log.4.gz ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log.4.gz"
fi
if [ -s /var/log/squid/access.log.3.gz ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log.3.gz"
fi
if [ -s /var/log/squid/access.log.2.gz ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log.2.gz"
fi
if [ -s /var/log/squid/access.log.1.gz ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log.1.gz"
fi
if [ -s /var/log/squid/access.log ]; then
    LOG_FILES="$LOG_FILES -l /var/log/squid/access.log"
fi

# Get yesterday's date
YESTERDAY=$(date --date "1 day ago" +%d/%m/%Y)
```

```
# Get 1 month ago date
MONTHAGO=$(date --date "1 month ago" +%d/%m/%Y)
```

```
exec /usr/bin/sarg \
    $LOG_FILES \
    -f /etc/sarg/sarg.conf \
    -o /var/www/html/sarg/monthly \
    -d $MONTHAGO-$YESTERDAY &>/dev/null
exit 0
```

```
# chmod 755 /etc/cron.daily/sarg /etc/cron.weekly/sarg /etc/cron.monthly/sarg
```

```
# mkdir /etc/sarg
# cp -r /usr/local/sarg/ /etc/
```

Edit file /etc/sarg/sarg.conf, dan sesuaikan option-option ini:

```
access_log /var/log/squid/access.log
date_format e
overwrite_report yes
```

Buat file konfigurasi apache untuk sarg:

```
# mkdir /var/www/sarg
# vi /etc/httpd/conf.d/sarg.conf
```

Dengan isi:

```
Alias /sarg /var/www/sarg
```

```
<Directory /var/www/sarg>
    DirectoryIndex index.html
    AllowOverride All
    Order deny,allow
    #Deny from all
    Allow from 127.0.0.1
    Allow from ::1
    # Allow from your-workstation.com
</Directory>
```

Dengan kita masukkan sarg ke dalam scheduler crontab, maka report-report sarg akan otomatis tercreate di /var/www/sarg setiap hari di jam 4.02am, setiap minggu jam 4.22am, dan setiap bulan jam 4.42am.

Kita dapat mengetestnya secara manual dengan menjalankan scriptnya, misalnya:

```
# /etc/cron.daily/sarg
```

Instalasi dan Konfigurasi Cacti

Sebelum menginstall, pastikan kita telah menginstall paket-paket ini:

1. snmp, net-snmp-utils (install dari CD Centos).
2. rrdtool dan perl-rrdtool (download dan install dari <http://dag.wieers.com/rpm/packages/rrdtool/>
3. # wget <http://dag.wieers.com/rpm/packages/rrdtool/perl-rrdtool-1.0.50-3.el4.rf.i386.rpm>
4. rpm -ivh perl-rrdtool-1.0.50-3.el4.rf.i386.rpm
5. # wget <http://dag.wieers.com/rpm/packages/rrdtool/rrdtool-1.0.50-3.el4.rf.i386.rpm>
6. rpm -ivh rrdtool-1.0.50-3.el4.rf.i386.rpm

Download cacti dari cacti.net:

```
# wget http://www.cacti.net/downloads/cacti-0.8.6j.tar.gz
```

Extract:

```
# tar zxvf cacti-0.8.6j.tar.gz
```

Move ke /var/www/cacti

```
# mv cacti-0.8.6j /var/www/cacti
```

Patch dahulu cactinya dengan mereplace file-file yang ada di /var/www/cacti/lib dengan yang kita download dari: <http://www.cacti.net/downloads/patches/0.8.6j/pre-patched/lib/>

Ada 3 buah file.

Siapkan snmpd:

Edit file /etc/snmp/snmpd.conf, cari baris ini:

```
# Make at least snmpwalk -v 1 localhost -c public system fast again.
#      name      incl/excl      subtree      mask(optional)
view   systemview   included      .1.3.6.1.2.1.1
view   systemview   included      .1.3.6.1.2.1.25.1.1
```

Ubah menjadi:

```
view   systemview   included      .1.3.6.1.2.1 <-- ini
view   systemview   included      .1.3.6.1.2.1.25.1.1
```

```
# chkconfig snmpd on
# service snmpd start
```

Create file /etc/httpd/conf.d/cacti.conf dengan isi:

```
Alias /cacti /var/www/cacti
<Directory /var/www/cacti>
```

```
        AllowOverride All
        DirectoryIndex index.php
        Order allow,deny
        Allow from all
</Directory>
```

Restart apache:

```
# service httpd restart
```

Siapkan mysql server:

```
# chkconfig mysqld on
# service mysqld start
```

Setup password root untuk mysql server:

```
# mysqladmin -u root password passwordygkamuinginkan
```

```
# cd /var/www/cacti
# mysqladmin -u root -p create cacti
# mysql -u root -p cacti < cacti.sql
# mysql -u root -p
mysql > GRANT ALL ON cacti.* TO cacti@localhost IDENTIFIED BY
'passwordygkamuinginkan';
mysql > quit
```

Edit file `/var/www/cacti/include/config.php`, sesuaikan settingannya dengan yang telah kamu setel (misalnya):

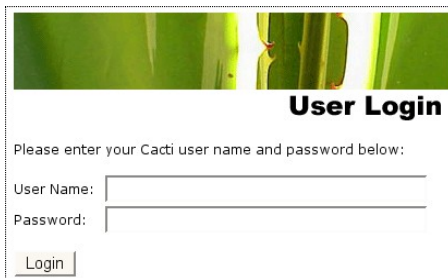
```
/* make sure these values reflect your actual database/host/user/password */
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cacti";
$database_password = "passwordygkamuinginkan";
```

Buka halaman awal cacti dari browser:

<http://localhost/cacti> (sesuaikan localhost dengan IP server kamu, misalnya <http://10.0.0.250/cacti>)

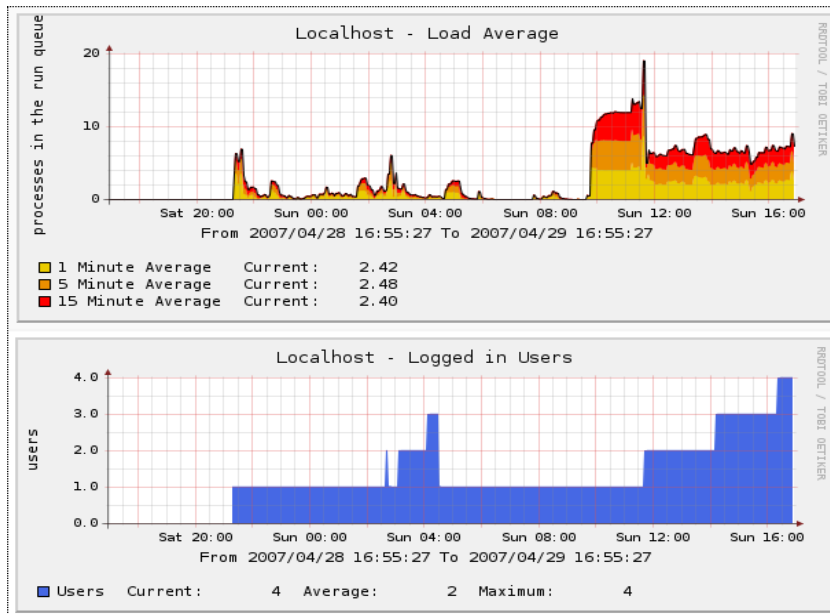
Akan muncul halaman installasi. Ikuti pertunjuknya, cuma ada 2 langkah, yang pertama welcome, kemudian halaman path-path binary yang dibutuhkan seperti snmpd, snmpdwalk, dll. Pastikan bahwa path-path itu telah tepat. Jika sudah, klik Finish.

Selanjutnya kita buka kembali halaman: <http://10.0.0.250/cacti>, maka akan muncul halaman login seperti di bawah ini. Username: admin, password: admin. Segera ganti passwordnya dengan yang baru dan baik.



Setelah login, kita harus mensetup cacti. Untuk detail lengkapnya mohon lihat di tutorialnya di www.cacti.net. Tapi secara garis besar, setupnya adalah sebagai berikut:

1. Klik devices
2. Add devices
3. Masukkan nama device, dan IP address dari device yang ingin kita monitor.
4. Pastikan bahwa cacti dapat mendetect devicenya dengan IP tersebut.
5. Kemudian masih dari menu device itu, kita klik Create Graph for this Host.
6. Pilih jenis-jenis data apa saja yang ingin kita grab dari device itu, misalnya Traffic IP, CPU load, dll. Klik Add di bawah graph yang kita inginkan.
7. Setelah itu masuk ke menu Graph Management.
8. Akan terlihat daftar graph yang tadi kita create.
9. Kita pilih graph yang kita inginkan, kemudian kita klik menu di bawah: Place on a Tree (Default Tree). Klik Add.
10. `chown -R apache.apache /var/www/cacti/log /var/www/cacti/rra`
11. Tambahkan ini di `/etc/crontab`:
12. `*/5 * * * * apache /usr/bin/php /var/www/cacti/poller.php > /dev/null 2>&1`
13. Test dengan menjalankan manual command di atas sebagai user apache:
14. `# su - apache`
15. `/usr/bin/php /var/www/cacti/poller.php`
16. Graphic akan mulai tercreate di directory `/var/www/cacti/rra` tersebut, dan kita dapat melihatnya dari <http://10.0.0.250/cacti/> di bagian graph seperti contoh di bawah ini:



Instalasi dan Konfigurasi DenyHosts

Download dari denyhosts.sourceforge.net

```
# wget http://optusnet.dl.sourceforge.net/sourceforge/denyhosts/DenyHosts-2.6-1.src.rpm
```

Compile:

```
# rpmbuild --rebuild DenyHosts-2.6.1.src.rpm
```

Install:

```
# rpm -ivh /usr/src/redhat/RPMS/noarch/DenyHosts-2.6-1.noarch.rpm
```

Copy template settingan dan daemonna:

```
# cp /usr/share/denyhosts/denyhosts.cfg-dist /etc/denyhosts.cfg
# cp /usr/share/denyhosts/daemon-control-dist /etc/init.d/denyhosts
# touch /var/log/denyhosts
```

Edit /etc/init.d/denyhosts:

```
DENYHOSTS_CFG = "/etc/denyhosts.cfg"
```

Masukkan denyhosts daemon ke dalam init script:

```
# chkconfig --add denyhosts
# chkconfig denyhosts on
```

Edit /etc/denyhosts.cfg, banyak sekali optionnya, tapi yang umum adalah ini:

```
# Redhat or Fedora Core:
```

```
SECURE_LOG = /var/log/secure
```

```
# Most operating systems:
```

```
HOSTS_DENY = /etc/hosts.deny
```

```
# never purge:
```

```
PURGE_DENY =
```

```
# To block only sshd:
```

```
BLOCK_SERVICE = sshd
```

```
# DENY_THRESHOLD_INVALID: block each host after the number of failed login
```

```
# attempts has exceeded this value. This value applies to invalid
```

```
# user login attempts (eg. non-existent user accounts)
```

```
#
```

```
DENY_THRESHOLD_INVALID = 3
```

```
# DENY_THRESHOLD_VALID: block each host after the number of failed
```

```
# login attempts has exceeded this value. This value applies to valid
```

```
# user login attempts (eg. user accounts that exist in /etc/passwd) except
# for the "root" user
#
DENY_THRESHOLD_VALID = 3

# DENY_THRESHOLD_ROOT: block each host after the number of failed
# login attempts has exceeded this value. This value applies to
# "root" user login attempts only.
#
DENY_THRESHOLD_ROOT = 1

ADMIN_EMAIL = root

# AGE_RESET_VALID: Specifies the period of time between failed login
# attempts that, when exceeded will result in the failed count for
# this host to be reset to 0. This value applies to login attempts
# to all valid users (those within /etc/passwd) with the
# exception of root. If not defined, this count will never
# be reset.
#
# See the comments in the PURGE_DENY section (above)
# for details on specifying this value or for complete details
# refer to: http://denyhosts.sourceforge.net/faq.html#timespec
#
AGE_RESET_VALID=5d

# AGE_RESET_ROOT: Specifies the period of time between failed login
# attempts that, when exceeded will result in the failed count for
# this host to be reset to 0. This value applies to all login
# attempts to the "root" user account. If not defined,
# this count will never be reset.
#
# See the comments in the PURGE_DENY section (above)
# for details on specifying this value or for complete details
# refer to: http://denyhosts.sourceforge.net/faq.html#timespec
#
AGE_RESET_ROOT=25d

# AGE_RESET_INVALID: Specifies the period of time between failed login
# attempts that, when exceeded will result in the failed count for
# this host to be reset to 0. This value applies to login attempts
# made to any invalid username (those that do not appear
# in /etc/passwd). If not defined, count will never be reset.
#
# See the comments in the PURGE_DENY section (above)
# for details on specifying this value or for complete details
# refer to: http://denyhosts.sourceforge.net/faq.html#timespec
#
AGE_RESET_INVALID=10d

# DAEMON_LOG: when DenyHosts is run in daemon mode (--daemon flag)
# this is the logfile that DenyHosts uses to report it's status.
```

```
# To disable logging, leave blank. (default is: /var/log/denyhosts)
#
DAEMON_LOG = /var/log/denyhosts
```

Test denyhosts:

```
# service denyhosts restart
```

Dari salah satu PC client coba lakukan login sebagai root, dan sengaja salahkan passwordnya. Monitor apa yang terjadi dengan `tail -f /var/log/messages` di gateway.

```
ssh root@10.0.0.250
```

Biarkan passwordnya salah.

Kita lihat di `tail -f /var/log/messages`

```
Apr 18 23:13:12 gateway sshd(pam_unix)[8048]: authentication failure; logname=
uid=0 euid=0 tty=ssh ruser= rhost=10.0.0.1 user=root
Apr 18 23:13:23 gateway sshd(pam_unix)[8048]: 2 more authentication failures;
logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.0.1 user=root
Apr 18 23:13:41 gateway sshd: refused connect from ::ffff:10.0.0.1
(::ffff:10.0.0.1)
```

Lihat di `/etc/hosts.deny`:

```
#
# hosts.deny      This file describes the names of the hosts which are
#                  *not* allowed to use the local INET services, as decided
#                  by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!

sshd: 10.0.0.1
```

Berarti sudah jalan.

Selamat! :)

Security Considerations

Ada baiknya bila website-website yang ada di gateway kita seperti web Sarg dan Cacti kita protek menggunakan password sehingga tidak sembarang orang dapat membacanya.

Caranya adalah buat file `.htaccess` di `/var/www/sarg` dan `/var/www/cacti`

Buat `.htaccess` di `/var/www/sarg`:

```
# vi /var/www/sarg/.htaccess
```

Dengan isi:

```
AuthUserFile /etc/httpd/conf/.passwd_sarg
AuthGroupFile /dev/null
AuthName "Tolong isi dahulu identitas dan password"
AuthType Basic
```

```
<LIMIT GET>
Order allow,deny
Require user admin
Allow from 127.0.0.1
Satisfy Any
</LIMIT>
```

Create passwordnya untuk web Sarg:

```
htpasswd -c /etc/httpd/conf/.passwd_sarg admin
```

Buat `.htaccess` di `/var/www/cacti`:

```
# vi /var/www/cacti/.htaccess
```

Dengan isi:

```
AuthUserFile /etc/httpd/conf/.passwd_cacti
AuthGroupFile /dev/null
AuthName "Tolong isi dahulu identitas dan password"
AuthType Basic
```

```
<LIMIT GET>
Order allow,deny
Require user admin
Allow from 127.0.0.1
Satisfy Any
</LIMIT>
```

Create passwordnya untuk web Cacti:

```
htpasswd -c /etc/httpd/conf/.passwd_cacti admin
```

Juga agar lebih secure lagi, kita atur agar orang tidak dapat login melalui ssh sebagai root. Jadi harus selalu user biasa dulu, baru setelah login bila ingin menjadi root dengan command `su -`.

Caranya adalah edit `/etc/ssh/sshd_config`, dan ubah option ini:

`PermitRootLogin yes` menjadi `PermitRootLogin no`.

Restart sshd:

```
# service sshd restart
```

Test dengan mencoba login dari PC client sebagai root. Pastikan bahwa sekarang sudah tidak bisa. Dan karena denyhosts sudah aktif, maka IP PC client itu akan terblock. Bila diinginkan dapat kita hapus di file `/etc/hosts.deny` sehingga dia dapat login kembali lewat ssh dari PC client itu.

Kesimpulan dan Penutup

Demikianlah Panduan Praktis dan Lengkap Membuat Internet Gateway Arinet kali ini. Dengan adanya gateway ini diharapkan kita semua dapat menikmati kehandalan Linux dan Software-software OpenSourcena untuk kepentingan gateway dan proxy.

Diharapkan hasil instalasi ini dapat menjadi starting point bagi kita untuk mulai mengeksplora dan belajar hal-hal baru lainnya di Linux, seperti web server, ftp server, mail server, dll.

Bagi perusahaan, lembaga, dllnya, diharapkan dengan adanya panduan ini dapat mengurangi penggunaan software-software bajakan atau menghemat biaya pembelian software proprietary yang bisa mencapai ratusan juta rupiah itu untuk keperluan yang lebih bermanfaat.

Penulis juga berharap bahwa panduan ini dapat menjadi inspirasi bagi rekan-rekan lainnya untuk dapat berbagi / sharing ilmu kepada komunitas dan bangsa kita, sehingga bila semakin banyak orang yang belajar dan menjadi pintar, maka ujung-ujungnya bangsa dan negara kita Indonesia pun akan dapat semakin maju.

Penulis menyadari bahwa panduan ini tidak sempurna dan banyak kekurangannya. Untuk itu mohon dimaafkan dan berikanlah saran/kritik kamu di Forum <http://linux2.arinet.org>

Akhir kata, selamat mencoba dan belajar.
Ilmu Pengetahuan adalah Milik Bersama.
Merdeka!

Fajar Priyanto

Seorang professional IT yang jatuh cinta kepada Linux.

<http://linux2.arinet.org>

ym: fajarpri

fajarpri@arinet.org

Changelog

1. 30 April 2007. Sedikit typo di `http_access our networks`, harusnya: **`http_access allow our_networks`**
2. 30 April 2007. Melengkapi script Sarg weekly dan monthly dengan `-f /etc/sarg/sarg.conf`.
3. 1 Mei 2007. Missing `/var/www/cacti/include/config.php` on cacti.