

Install OpenSBC on Vyatta Firewall

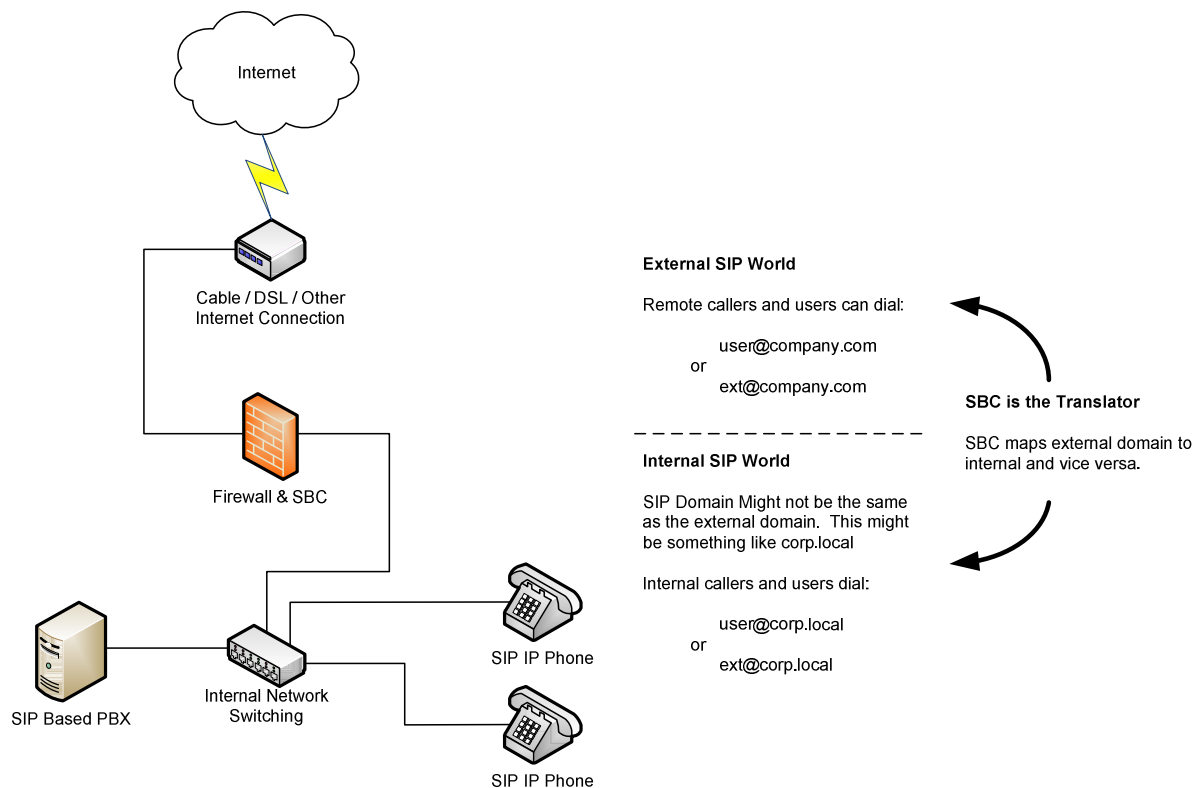
By: Michael W. Picher

Date: January 2009

Introduction

The following instructions will help you create a fully functional SIP Session Border Controller. Utilizing the Open Source SBC known as OpenSBC (www.opensourcesip.org) and the Open Source and high performance firewall known as Vyatta (www.vyatta.org) you will be able to create a single server solution. This solution was designed to work with the Open Source PBX sipX Enterprise Communications Server (sipXecs) but should function for most pure SIP solutions with minor adjustments.

A Session Border Controller is a device used in VoIP networks to provide control over the signaling and sometimes also the media streams involved in setting up, conducting, and tearing down calls. The battle with SIP traffic is usually in dealing with Network Address Translation (NAT) issues. Whether on the PBX side or on the far end for a teleworker. The SBC helps solve those problems.



The Session Border Controller also allows administrators to not have to expose their PBX equipment directly to the Internet. The SBC acts as a buffer between the two worlds and may have some security measures implemented.

Additionally it is desirable to be able to prioritize outbound Internet traffic through a single interface to the Internet. If a Session Border Controller is placed in parallel with a corporate firewall they will both contend for the same bandwidth. With the SBC and firewall in line the voice traffic can be prioritized properly over the data traffic.

OpenSBC can do much more than manage SIP traffic in and out of a network. Another great feature of the product is the ability to setup trunks to Internet Telephony Service Providers (ITSP's). This document however is concerned with SIP and NAT Traversal issues, so let's get on with it.

Install Vyatta

This work was done with Vyatta 5 Beta.

Download the Vyatta LiveCD ISO from <http://www.vyatta.org/downloads>

Burn the ISO to a CD (imgburn (www.imgburn.com) works great)

1. Boot from Vyatta LiveCD ISO. Press ENTER.
2. At login, username 'root', password 'vyatta'.
3. Enter 'install-system'
4. Configure as needed... (see example configuration at the end of this document)

Download/Install Items Needed for OpenSBC Compliation

This is kind of a kitchen sink approach as there are probably more packaged listed than needed.

1. Login to Firewall as user 'vyatta'
2. `cd /etc/apt`
3. `su`
4. Password: (enter root password)
5. `nano -w sources.list`
6. Add line: "`deb ftp://ftp.us.debian.org/debian/ lenny main contrib non-free`"

7. Ctrl-X and Y to overwrite
8. apt-get update
9. apt-get install -y mc autoconf automake cvs flex expat libexpat1-dev libtool build-essential libxml2 libxml2-dev libtiff4 libtiff4-dev php5 php5-cli php5-mysql php5 php5-cli php5-mysql php5-gd mysql-server libmysqlclient15-dev php-pear php-db curl sox apache2 libssl-dev libncurses5-dev bison libaudiofile-dev subversion libnewt-dev libcurl3-dev libnet-ssleay-perl openssl ssl-cert libauthen-pam-perl libio-pty-perl libmd5-perl libpg-perl libdbd-pg-perl php5-pgsql sqlite3 libsqlite3-dev openssl ssl-cert libapache2-mod-php5 php5-cli php5-common phpMyAdmin php5-mcrypt mcrypt phppgadmin apache2 libmcrypt-dev

Get OpenSipStack and OpenSBC from CVS:

This will download the most current OpenSIPStack and OpenSBC code from the development area.

1. `cd /usr/src`
2. `cvs -d:pserver:anonymous@opensipstack.cvs.sourceforge.net:/cvsroot/opensipstack login`
3. `cvs -z3 -d:pserver:anonymous@opensipstack.cvs.sourceforge.net:/cvsroot/opensipstack co -P opensipstack`
4. `cvs -z3 -d:pserver:anonymous@opensipstack.cvs.sourceforge.net:/cvsroot/opensipstack co -P opensbc`

Compile / Make OpenSipStack and OpebSBC:

The following will compile OpenSIPStack and OpenSBC

1. `cd /usr/src/opensipstack`
2. `chmod +x ./configure`
3. `./configure`
4. `make bothnoshared`
5. `cd ../opensbc`
6. `chmod +x ./configure`
7. `./configure`
8. `make bothnoshared`
9. `make distrib`

Relocate Executables

Copy the executables to a more acceptable location on the server.

1. `cp /usr/src/opensbc/distrib/* /usr/local/bin`

Fix Shell Scripts

The following will modify the startup and shutdown shell scripts.

1. Edit `/usr/local/bin/startup.sh` (i use 'nano -w /usr/local/bin/startup.sh')
2. Modify the startup command to: `./opensbc -d -p /var/run/opensbc.pid -H 65536 -C 1024000`
3. Modify the shutdown command to: `./opensbc -k -p /var/run/opensbc.pid`

Make OpenSBC run at Startup

The following will make OpenSBC run when the Linux machine starts up.

1. copy `startup.sh` to `/etc/init.d` (`cp /usr/local/bin/startup.sh /etc/init.d/opensbc.sh`)
2. Modify `opensbc.sh` to make sure it runs as root and so it can find the application.
 - a. `nano -w /etc/init.d/opensbc.sh`
 - b. Make the line read: `/usr/local/bin/opensbc -u root -d -p /var/run/opensbc.pid -H 65536 -C 1024000`

Note: In Debian the startup programs run from the `rc2.d` directory, begin with cap S and the order number it should start in. Create a symbolic link in that directory to the `opensbc.sh` script file.

3. `ln -fs /etc/init.d/opensbc.sh /etc/rc2.d/S92opebsbc`

Miscellaneous Information

Log / Ini Files

The ini file is in: /root/OpenSIPStack/OpenSBC_data

The log files are in /root/OpebSIPStack/OpebSBC_data/logs

To watch a log file use tail -f

```
tail -f /root/OpenSIPStack/OpebSBC_data/logs/b2bua-....
```

To get to OpenSBC Administration Console

<http://inside.ip.addr.vyatta:9999>

OpenSBC.ini file:

See the important settings in **RED**.

[Solegy]

RTTS-Client-Address=(gets populated automatically with outside IP, this is of no use)

[OpenSBC-General-Parameters]

SIP-Log-Level=1

PTRACE-Log-Level=1

Log-File-Prefix=b2bua

SBC-Application-Mode=**B2BUpperReg Mode**

Interface-Address Array Size=0

Enable-Backdoor-Port=True

Enable-Trunk-Port=True

Enable-Calea-Port=True

RTP-Min-Port=**10000**

RTP-Max-Port=**20000**

Enable-Local-Refer=False

Max-Forwards=70

Encryption-Mode=XOR

Encryption-Key=GS

Transaction-Thread-Count=10

Session-Thread-Count=10

Alerting-Timeout=30000

Seize-Timeout=60000

SIP-Timer-B=Default

SIP-Timer-H=Default

Session-Keep-Alive=1800

Session-Max-Life-Span=10800

Max-Concurrent-Session=100

Max-Call-Rate-Per-Second=10

[Upper-Registration]

Enable-Stateful-Reg=False

Rewrite-TO-Domain=True

Rewrite-FROM-Domain=True

Route-List Array Size=1

Route-List 1=[sip:*@externalsip.domain] sip:internalsip.domain;domain=internalsip.domain

[Internal-DNS-Mapping]

Internal-DNS-Map Array Size=2

Internal-DNS-Map 1=[sip:internalsip.domain] sip:internal.ip.of.sipxpbx:5060

Internal-DNS-Map 2=[sip:internal.fqdn.ofsipxpbx] sip:internal.ip.of.sipxpbx:5060

[Proxy-Relay-Routes]

Drop-Routes-On-Ping-Timeout=False

Proxy-Resolve-To-URI=True

Route-List Array Size=0

[B2BUA-Routes]

Enable-Route-Scripting=False

Route-Script=b2bua-route.cscript

Route-List Array Size=1

Route-List 1= [sip:*@externalsip.domain] sip:internalsip.domain

Insert-Route-Header=True

Rewrite-TO-URI=True

Prepend-ISUP-OLI=False

Route-By-Request-URI=True

Route-By-To-URI=False

Drop-Routes-On-Ping-Timeout=False

Use-External-XML=False

External-XML-File=b2bua-route.xml

[Media-Server]

Enable-Media-Server=False

Media-Server-Number=5000

Codec-List Array Size=0

No-RTP-Proxy-On-All-Transfers=False

Enable-Announcement-Service=False

4xx-Error-Map=prompts/basic/cant_complete.wav

5xx-Error-Map=prompts/basic/cant_complete.wav

6xx-Error-Map=prompts/basic/cant_complete.wav

Announcement-Error-Map Array Size=0

[Outbound-Proxies]

Outbound-Proxies Array Size=0

[Local-Domain-Accounts]

Accept-All-Registration=True

Account-List Array Size=0

A Simple Vyatta Configuration

```
firewall {  
    broadcast-ping disable  
    name ALLOW_ESTABLISHED {  
        rule 10 {  
            action accept  
            state {  
                established enable  
            }  
        }  
    }  
    name INBOUND {  
        rule 20 {  
            action accept  
            destination {  
            }  
            log disable  
            state {  
                established enable  
                related enable  
            }  
        }  
        rule 999 {  
            action drop  
            destination {
```

```
        address 0.0.0.0/0
    }
    log enable
    source {
        address 0.0.0.0/0
    }
}
:
name OUTBOUND {
    rule 999 {
        action accept
        destination {
            address 0.0.0.0/0
        }
        log disable
        source {
            address 0.0.0.0/0
        }
    }
}
name TO-ROUTER {
    rule 10 {
        action accept
        log disable
        state {
```

```
        established enable
        related enable
    }
}
rule 30 {
    action accept
    icmp {
        type 3
    }
    log disable
    protocol icmp
}
rule 32 {
    action accept
    icmp {
:       type 8
    }
    log disable
    protocol icmp
}
rule 34 {
    action accept
    icmp {
        type 11
    }
}
```

```
log disable
protocol icmp
}
rule 40 {                                     (Allow SIP Signalling Traffic In)
    action accept
    destination {
        port 5060
    }
    log enable
    protocol udp
}
rule 41 {                                     (Allow RTP (voice) traffic in)
    action accept
    destination {
        port 10000-20000
    }
    log enable
    protocol udp
}
rule 999 {
    action drop
    destination {
        address 0.0.0.0/0
    }
    log enable
```

```
    source {
        address 0.0.0.0/0
    }
}
}
```

```
interfaces {
```

```
    ethernet eth0 {
```

```
        address 172.16.1.254/24
```

```
        description inside
```

```
        firewall {
```

```
            in {
```

```
                name OUTBOUND
```

```
            }
```

```
        }
```

```
        hw-id {auto populated with inside MAC addr}
```

```
        mtu 1500
```

```
    }
```

```
    ethernet eth1 {
```

```
        address dhcp                                (I'm getting my IP DHCP from cable modem)
```

```
        firewall {
```

```
            in {
```

```
                name INBOUND
```

```
        :
```

```
    }
```

```
    local {
```

```

    name TO-ROUTER
  }
}
hw-id {auto populated with outside MAC addr}
}
loopback lo {
}
}
service {
  dhcp-server {
    disabled false
    shared-network-name ETH0_POOL {
      authoritative disable
      subnet 172.16.1.0/24 {
        (Inside Network IP Range)
        default-router 172.16.1.254 (Inside IP Address of Vyatta)
        dns-server 172.16.1.2 (IP Address of PBX or internal DNS Server)
        dns-server 208.67.222.222 (Extra DNS Servers...)
        dns-server 208.67.220.220
        domain-name xyzcompany.com (Internal SIP domain)
        ntp-server 172.16.1.2 (To Get Time from PBX)
        start 172.16.1.100 {
          stop 172.16.1.200
        }
        tftp-server-name 172.16.1.2 (To Download Phone Config Files from PBX)
      }
    }
  }
}

```

```
}  
}  
dns {          (If you are using dynamic DNS for Outside Interface)  
  dynamic {  
    interface eth1 {  
      service dyndns {  
        host-name sipxecs.dyndns.info  
        login LoginID  
        password password  
      }  
    }  
  }  
}  
nat {  
  rule 1 {  
    outbound-interface eth1  
    type masquerade  
  }  
}  
ssh {  
  allow-root true  
}  
webproxy {  
  cache-size 500  
  listen-address 172.16.1.254 {
```

```
    }
  }
:
}
system {
  domain-name xyzcompany.com

  host-name fw1

  login {
    user root {
      authentication {
        encrypted-password $1$0/Nwe.bk$rM/D4fTAHbvjaLarha/xK/
      }
    }
  }

  user vyatta {
    authentication {
      encrypted-password $1$6xFKV1OQ$ifrekBjneusx1kacuaJm8/
    }
  }
}

ntp-server 69.59.150.135

package {
  auto-sync 1

  repository community {
    components main

    distribution stable

    url http://packages.vyatta.com/vyatta
```

```
    }  
  }  
  syslog {  
    console {  
      facility all {  
        level err  
      }  
    }  
    host 172.16.1.129 {  
      facility all {  
        level err  
      }  
      facility kern {  
        level debug  
      }  
      facility local0 {  
        level debug  
      }  
    }  
  }  
}
```