Select a page

- Blogs
- Contact
- Donate to ISC
- Shop
  - Cart
- Customer Login

- Twitter
- Facebook
- LinkedIn
- YouTube

Internet Systems Consortium Internet Systems Consortium

Select a page

- Software
  - BIND
    - Documentation
      - BIND 9.10 ARM
      - BIND 9.9 ARM
      - BIND 9.8 ARM
      - BIND 9.6 ESV ARM
      - DNSSEC Guide
    - BIND Users searchable archive
    - Report a Bug
    - DNSSEC
    - ISC's DLV Registry
    - bind.keys
  - ISC DHCP
    - DHCP-users searchable archive
    - Report a Bug
  - Kea DHCP server
  - Other Software
    - DS-Lite
    - AFTR
    - OHGF
    - OSRF
    - BIND 10
    - Libbind
  - Affiliated Projects
  - OpenPGP Key
  - Software Support Policy
    - Version Numbering
    - ISC License
  - Security Advisory
    - BIND 9 Vulnerability Matrix
- Support
  - BIND Support Subscription
  - BIND Documentation
    - BIND 9.10 ARM
      - BIND 9.10 New Features
    - BIND 9.9 ARM
    - BIND 9.8 ARM
    - BIND 9.6 ESV ARM
    - BIND DNSSEC Guide
  - DHCP Support Subscription
    - DHCP 4.3.1 Documentation
  - Knowledge Base
  - Report a Bug
  - Security Advisory
    - BIND9 Vulnerability Matrix
- Community
  - Blogs

    - Certification
    - Conferences
    - Mailing Lists
    - Report a Bug
    - RFCs
        - RFCs by ISC
        - DNS RFC
    - Source Repository
        - ISC Contributors Guide
        - BIND Developer Information
        - BIND 9 Coding Style
        - BIND Contributors Guide
        - Kea Developer Guide
        - Kea Wiki
    - Tools and Resources
    - Webinars and other Presentations
- Network
    - F-Root
        - Network Peering
        - F-Root Resources
    - SNS-PB
    - Hosted@ISC
    - ISC Domain Survey
- About Us
    - 2014 Annual Report
    - ISC Team
    - Board of Directors
    - Press Release
    - ISC's Value
    - ISC Timeline
    - ISC Acceptable Use Policy
    - Jobs
    - Contact
- Professional Support
    - BIND Subscription
        - BIND subscription contact
    - DHCP Subscription
        - DHCP subscription contact
    - Unbound & NSD Support Subscription

ISC Internet Systems Consortium

ISC Internet Systems Consortium

- Software
    - BIND
        - Documentation
            - BIND 9.10 ARM
            - BIND 9.9 ARM
            - BIND 9.8 ARM
            - BIND 9.6 ESV ARM
            - DNSSEC Guide
        - BIND Users searchable archive
        - Report a Bug
        - DNSSEC
        - ISC's DLV Registry
        - bind.keys
    - ISC DHCP
        - DHCP-users searchable archive
        - Report a Bug
    - Kea DHCP server
    - Other Software
        - DS-Lite
        - AFTR

- OHGF
- OSRF
- BIND 10
- Libbind
- Affiliated Projects
- OpenPGP Key
- Software Support Policy
  - Version Numbering
  - ISC License
- Security Advisory
  - BIND 9 Vulnerability Matrix

- Support
  - BIND Support Subscription
  - BIND Documentation
    - BIND 9.10 ARM
      - BIND 9.10 New Features
    - BIND 9.9 ARM
    - BIND 9.8 ARM
    - BIND 9.6 ESV ARM
    - BIND DNSSEC Guide
  - DHCP Support Subscription
    - DHCP 4.3.1 Documentation
  - Knowledge Base
  - Report a Bug
  - Security Advisory
    - BIND9 Vulnerability Matrix
- Community
  - Blogs
  - Certification
  - Conferences
  - Mailing Lists
  - Report a Bug
  - RFCs
    - RFCs by ISC
    - DNS RFC
  - Source Repository
    - ISC Contributors Guide
    - BIND Developer Information
    - BIND 9 Coding Style
    - BIND Contributors Guide
    - Kea Developer Guide
    - Kea Wiki
  - Tools and Resources
  - Webinars and other Presentations
- Network
  - F-Root
    - Network Peering
    - F-Root Resources
  - SNS-PB
  - Hosted@ISC
  - ISC Domain Survey
- About Us
  - 2014 Annual Report
  - ISC Team
  - Board of Directors
  - Press Release
  - ISC's Value
  - ISC Timeline
  - ISC Acceptable Use Policy
  - Jobs
  - Contact
- Professional Support
  - BIND Subscription
    - BIND subscription contact
  - DHCP Subscription

Lightweight DS-Lite Setup Guide

Requirements

General

- Previous knowledge on IPv4 and IPv6, tunneling techniques, Linux, routing and dynamic host configuration protocol (DHCP), are highly recommended;
- An individual with systems administration background should be able to perform these tasks and diagnose and potential problems using standard Unix and network tools.

Hardware Requirements

There are at least two possible modes to setup the system: on a Virtual Machine environment, and on a real hardware.

Virtual Machine Setup

- Host computer capable of running at least three (3) virtual machines at the same time.
- Standard setup with at least 8 GB of RAM, CPU with support of hardware virtualization (must be enabled in the BIOS) and at least one network interface.
- Virtualization software: VMWare Workstation (>=8.0) or VirtualBox (>=4.1.x) (others not tested but could work).

Real Hardware Setup

- One (or more) computers for the SD-AFTR with at least two network interfaces (eth0 for WAN, eth1 for transport network);
- One (or more) computers (or CPEs) for the SD-B4s with at least two network interfaces;
- At least one gigabit network switch (24-port) with VLAN support and port mirroring (if performing packet analysis).

Network Requirements

- A public IP address for each of the SD-AFTR;
- A public subnet routed (statically or dynamically) to the SD-AFTRs;
- An global IPv6 prefix (/48) if Dual-Stack is wanted;
- A private IPv6 transport network (2001:db8::/32);
- As many HOME-LAN prefixes as needed (can be the same across all networks).

Software Requirements

In order to get the software packages ready for deployment, some compiling work will need to happen, either in the component itself, in another computer with the same architecture and OS version, or via a cross compiler. The following software packages are recommended for debian/Ubuntu systems (installed via: aptitude install <package-name>):

- build-essential – Informational list of build-essential packages
- autoconf – automatic configure script builder
- libssl-dev – SSL development libraries (needed to build DHCP and BIND)
- bzip2 – file compressor/decompressor
- libtool
- libpcap-dev
- lsof
- others as required by compilation processes…

On the SD-AFTR

- aftr: is a patched aftr-1.1 distribution that supports stateless NAT using port ranges that are forwarded according to an algorithm computed as packets arrive to the box;
- dhcpd (TSV mode): a patched version of ISC DHCP-4.2.0 that supports TSV (Transport Services) which is a DHCPv4 server that is capable of answering IPv4 requests over IPv6;
- dhcpd (IPv6 mode): a standard ISC DHCP 4.x running in IPv6 mode.
- bind9: a standard ISC-BIND distribution capable of answering recursive queries for both IPv6 and IPv4 clients;
- linux: debian, Ubuntu, fedora, or any other distribution of choice.

On the SD-B4

- dhclient (IPv6 mode): ISC-DHCP 4.2.0 that supports setting the type of DUID to use (-D LL). This will be used to configure the IPv6 address of the WAN interface on the SD-B4, including the IPv6 default gateway and name servers;

- dhclient (IPv4 mode): ISC DHCP 4.x client used to configuring the IPv4 (shared address) on the WAN interface and the port ranges to be used by the client.
- dhccra: included in the same patched version of ISC-DHCP 4.2.0, is a DHCP Client Relay Agent (cra), that listens for DHCPv4 (ipv4) requests and forwards them to the DHCP (tsv) server running in the AFTR;
- dhcpd: a standard version of the ISC DHCP 4.x to configure clients on the local network (clients).
- sdctld: a “Stateless Deterministic Control Daemon”, which listens on the local network for NAT-PMP, UPNP and PCP requests for port redirections.
- prmon: a PCAP-based ICMP type 3 code 13 that is capable of rebooting the SD-B4 in case the port range has changed for this particular customer.

On the Client

Clients behind the SD-B4 are standard computers and devices that request their network configuration via DHCPv4. For the purpose of this setup, it would be important to have a web browser installed, as well as any other programs that use TCP/UDP/ICMP protocols.

Add a service such as ssh-server.

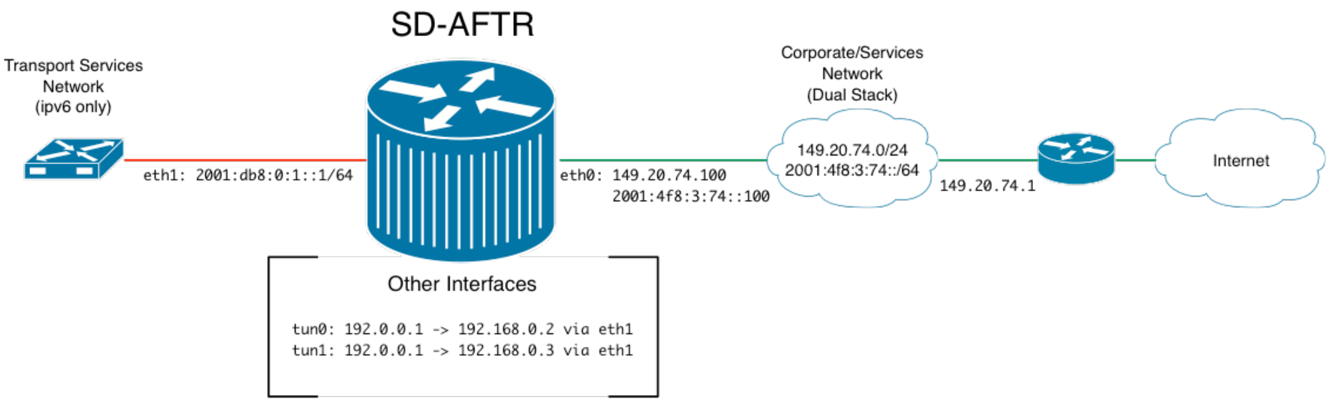
Software Packages

The software packages used in this setup are not available for general use yet. They remain part of our experimental branches to address the SDNAT proposal as a reference implementation.

The software package necessary to test the SDNAT setup includes:

File	Target Component	Description
DHCP_rt28195a.tbz	SD-B4 and SD-AFTR	Patched version of DHCP-4.2.0 which includes: • dhclient: DHCP Client with DUID setting (sdb4); • dhccra: Dynamic Host Client Relay Agent (sdb4) • dhcpd: DHCP server with TSV (Transport Services) support. (sdaftr) • dhctra: DHCPv4 IPv6 Transport Relay Agent (currently not used)
rt28354.tbz	SD-B4 and SD-AFTR	Includes multiple programs and sample configurations used to implement SDNAT features: • aftr: patched version of after that supports stateless mode (sdaftr). • sdctld: stateless deterministic control, a miniupnpd implementation that supports stateless deterministic mode for NAT-PMP, UPnP IGD and PCP (sdb4). • conf/testbeds/hw/*: various sample configuration files used both in the sdaftr. • conf/sdb4-ubuntu/*: various sample configurations for the SD-B4.

SD-AFTR Setup



1. Perform an OS installation with standard system packages (Debian, Ubuntu or Fedora) vanilla install.

2. Configure the network interfaces (debian and Ubuntu):

a. Edit /etc/network/interfaces and include:

```
[sourcecode language="powershell"] auto eth0 eth1

# Upstream network

iface eth0 inet static
address 149.20.74.100
netmask 255.255.255.0
gateway 149.20.64.1

iface eth0 inet6 static
address 2001:4f8:3:74::100
netmask 64
gateway 2001:4f8:3:74::1

# Transport Services Network
iface eth1 inet6 static
address 2001:db8:0:1::1
netmask 64
[/sourcecode]
```

Tunnel interfaces are created when the aftr daemon boots.

b. Restart the network interfaces by issuing the following command (as root): /etc/init.d/networking restart

3. Make sure that the organization providing IP connectivity properly routes a /28 of IPv4 space either static or dynamically to the SDAFTR device(s). This network will be used as a shared pool for SDNAT clients. For now lets assume the network assigned is: 149.20.20.0/28

4. Unpack the rt28354.tbz file and compile the aftr binary:

```
[sourcecode language="powershell"] $ tar –jxvf /path/to/rt28354.tbz
$ cd rt28354
$ ./configure
$ make
$ cp aftr /usr/local/sbin
[/sourcecode]
```

5. Copy the sample aftr configuration located in rt28354/conf/testbeds/hw/aftr-sdaftr1.conf to /etc/aftr/sdaftr.conf.

6. Copy the aftr configuration script located in rt28354/conf/testbeds/hw/aftr-sdaftr1-script to /etc/aftr/aftr-sdaftr-script.

7. Edit the /etc/aftr/sdaftr.conf to reflect the local setup, in our example, the configuration file would look like this:

```
[sourcecode language="powershell"] default tunnel nat tcp maxcount 2000
default tunnel nat udp maxcount 2000
default tunnel nat icmp maxcount 2000
default pcpd stateless
default private 149.20.20.0/28

## section 1: required parameters

acl6 2001:db8::/32
address endpoint 2001:db8::1
address icmp 149.20.20.1
pool 149.20.20.2 tcp 50000-59999
pool 149.20.20.2 udp 50000-59999
pool 149.20.20.2 echo 50000-59999
```

```
## section 2: reloadable parameters
prr 2001:db8:0:1::2 tcp 149.20.20.2 1024
prr 2001:db8:0:1::2 tcp 149.20.20.2 1025
prr 2001:db8:0:1::2 tcp 149.20.20.2 1026
prr 2001:db8:0:1::2 tcp 149.20.20.2 1027
prr 2001:db8:0:1::2 tcp 149.20.20.2 1028
prr 2001:db8:0:1::2 tcp 149.20.20.2 1029
prr 2001:db8:0:1::2 tcp 149.20.20.2 1030
prr 2001:db8:0:1::2 tcp 149.20.20.2 1031
prr 2001:db8:0:1::2 tcp 149.20.20.2 1032
… continued (see original file)
[/sourcecode]
```

> The file is much bigger than what is shown here. Do a global search and replace of the ip address 149.20.74.65 with the IP to be shared among multiple clients. The format for the prr command is: prr <client_ipv6_address> <proto> <shared_ip_address> <port>. This configuration is usually generated from a script.
>
> Note the change on the default private and icmp source as well.

8. Edit the /etc/aftr/aftr-sdaftr-script file and update the CIDR block referenced in the ‘ip route’ command with the IP range to be shared. Note the change on the IP addresses more specifically (highlighted):

```
[sourcecode language=”powershell” highlight=”26,27″] #!/bin/sh

# Copyright (C) 2012 Internet Systems Consortium, Inc. ("ISC")
#
# Permission to use, copy, modify, and/or distribute this software for any
# purpose with or without fee is hereby granted, provided that the above
# copyright notice and this permission notice appear in all copies.
#
# THE SOFTWARE IS PROVIDED "AS IS" AND ISC DISCLAIMS ALL WARRANTIES WITH
# REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
# AND FITNESS. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, DIRECT,
# INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
# LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
# OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
# PERFORMANCE OF THIS SOFTWARE.

# $Id: 915470b80725007ae12f5484d9905f14ecfe761b $

aftr_start()
{
sysctl -w net.ipv4.ip_forward=1
sysctl -w net.ipv6.conf.all.forwarding=1
sysctl -w net.ipv4.conf.all.rp_filter=0

ip link set tun0 up
ip addr add 192.0.0.1 peer 192.0.0.2 dev tun0
ip route add 149.20.20.0/28 dev tun0
ip -6 route add 2001:db8::/64 dev tun0
}

aftr_stop()
{
ip link set tun0 down
}

set -x

case "$1" in
start)
aftr_start
;;
stop)
aftr_stop
;;
*)
echo "Usage: $0 start|stop"
exit 1
;;
esac

exit 0
[/sourcecode]
```

9. Run ‘aftr’ by executing the command: aftr -c /etc/aftr/aftr-sdaftr1.conf -s /etc/aftr/aftr-sdaftr1-script
Note: since this is the first time it will be running, it is recommended to use the –goption to prevent aftr from detaching itself from the terminal and going into background mode. More info on the AFTR-1.1 manual

10. If you want the process to start when the server boots up, add the command to /etc/rc.local (before the exit option)

DHCPv6 for customers (WAN)

This DHCP server assigns IPv6 addresses as well as standard network parameters to clients in the transport network. Since we're not using any non-standard options at this level, we can use the ISC-DHCP server supplied by the OS distribution, however, to avoid complexity and having multiple DHCP servers installed on the system, the TSV can be used to serve IPv6 requests as well.

1. Use the file: rt28354/conf/testbeds/hw/dhcpd6-sdaftr1.conf as the main configuration file for the DHCPv6 (ISC 4.x) server. (copy to /usr/local/etc/dhcpd6.conf)
2. Edit the DHCPv6 configuration file /usr/local/etc/dhcpd6.conf file and update the host-identifier option with the correct mac addresses of the clients (B4s). Note that the mac addresses require a: 00:03:00:01: prepended. For example, a host with a mac address of: "08:00:27:07:cf:d3" will use a host identifier of: "host-identifier option dhcp6.client-id 00:03:00:01:08:00:27:07:cf:d3;"
3. ISC's DHCP server requires a lease file file to store its leases. The file must exist (as an empty file) before starting up the server.
4. Start up the dhcp server by executing: dhcpd –q -6 –cf /usr/local/etc/dhcpd6.conf –lf /tmp/dhcpv6.leases eth1
5. If needed, inspect with the command netstat –ap (as root) (or lsof) to see if the service is running and waiting for DHCPv6 requests (scroll to right to see the rest of the code):

```
[sourcecode language=”powershell”] udp6 0 0 [::]:dhcpv6-server [::]:* 979/dhcpd
raw 0 0 *:icmp *:* 7 979/dhcpd
[/sourcecode]
```

DHCP for Transport Services Network (TSV)

This instance of DHCP server is a patched version of ISC-DHCP v4.2.0 with a special support mode (-tsv), which enables the server to answer to IPv4 requests over an IPv6 transport connection.

The TSV dhcp server also provisions clients with the SD parameters (port ranges) necessary for SDNAT operation.

For this setup, it will be necessary to use the version included in the file DHCP_rt28195a.tbz as follow:

1. Unpack the distribution by issuing the command: tar jxvf /path/to/DHCP_rt28195a.tbz

2. Since experimental versions of DHCP do not carry BIND with them, it is necessary to obtain a copy of the 'bind/' subdirectory from a stock version from ISC's website. To do so, download: http://ftp.isc.org/isc/dhcp/dhcp-4.2.0-P2.tar.gz and copy the 'bind/' subdirectory into the DHCP_rt28195a/ subdirectory. A sample way of doing this would be:

```
[sourcecode language=”powershell”] # wget http://ftp.isc.org/isc/dhcp/dhcp-4.2.0-P2.tar.gz
# tar zxvf dhcp-4.2.0-P2.tar.gz dhcp-4.2.0-P2/bind/
# cp -ar dhcp-4.2.0-P2/bind/ DHCP_rt28195a/
[/sourcecode]
```

3. Build and install this version of DHCP into /usr/local a sample procedure would be:

```
[sourcecode language=”powershell”] # cd DHCP_rt28195a
# ./configure –prefix=/usr/local
# make 1>make.log 2>&1

don't forget to inspect make.log for errors

# make install
[/sourcecode]
```

4. Copy the rt28354/conf/testbeds/hw/tsv-sdaftr1.conf file and place it into /usr/local/etc/tsv.conf.

5. Edit the tsv.conf file and make sure you include the proper IP and MAC addresses for the host declarations.

6. A sample working tsv.conf file would look like:

```
[sourcecode language=”powershell”] option port-range-min code 240 = unsigned integer 16;
option port-range-max code 241 = unsigned integer 16;

allow unknown-clients;
default-lease-time 1800;
max-lease-time 7200;

shared-network internal {
subnet 149.20.20.0 netmask 255.255.255.240 {
}
subnet6 2001:db8:0:1::0/64 { }
}

host sdb4_1 {
hardware ethernet 74:44:01:93:67:1b;
fixed-address 149.20.20.2;
option port-range-min 1024;
option port-range-max 1535;
}
```

```
host sdb4_2 {
hardware ethernet c4:3d:c7:9d:df:1e;
fixed-address 149.20.20.2;
option port-range-min 2048;
option port-range-max 2559;
}
[/sourcecode]
```

7. Create a lease file for the TSV dhcp server by issuing: touch /tmp/tsv.leases

8. Start up the service: /usr/local/sbin/dhcpd -tsv -cf /usr/local/etc/tsv.conf -lf /tmp/tsv.leases

9. As with the previous DHCP setup, look in /var/log/syslog and /var/log/daemon.log for errors as well as in netstat –ap or lsof to make sure that the server is listening for DHCPv4 requests over IPv6.
Sample output of the netstat –ap command with a TSV DHCP server should look like:

```
[sourcecode language=”powershell”] udp 0 0 *:40289 *:* 1218/dhcpd
udp6 0 0 [::]:bootps [::]:* 1218/dhcpd
udp6 0 0 [::]:63061 [::]:* 1218/dhcpd
raw 0 0 *:icmp *:* 7 1218/dhcpd
[/sourcecode]
```

10. To make sure the service starts every time the server boots up, include the command in the /etc/rc.localfile

Recursive DNS Server Setup

This last component is needed to provide clients with means to resolve DNS queries. Since there is nothing special to this setup, it is possible to use the BIND version (or any other vendor’s) included in your linux distribution.

Sample steps to setup a recursive DNS server in debian are:

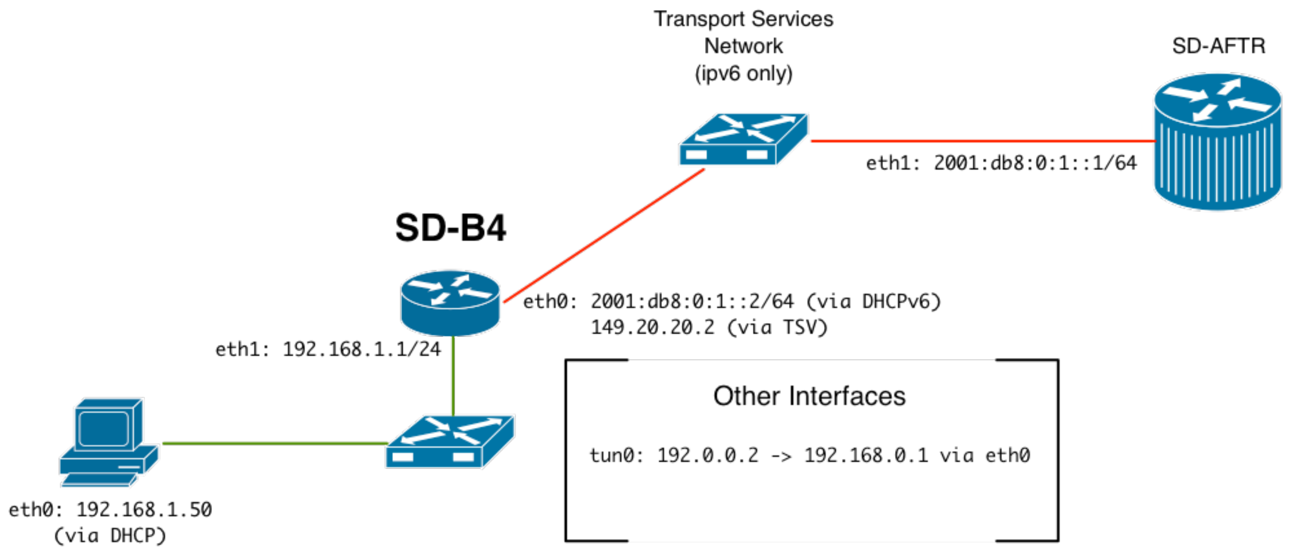
1. Install the package: aptitude install bind9 bind9utils
2. Modify the /etc/named/named.conf.options file and include the configuration to allow recursion for your networks:

   ```
   [sourcecode language=”powershell”] acl mynetworks {
   140.20.20.0/28;
   2001:db8::/32;
   127.0.0.1/32;
   ::1/128;
   };
   options {
   directory "/var/cache/bind";
   // If there is a firewall between you and nameservers you want
   // to talk to, you may need to fix the firewall to allow multiple
   // ports to talk. See http://www.kb.cert.org/vuls/id/800113

   // If your ISP provided one or more IP addresses for stable
   // nameservers, you probably want to use them as forwarders.
   // Uncomment the following block, and insert the addresses replacing
   // the all-0’s placeholder.

   // forwarders {
   // 0.0.0.0;
   // };

   auth-nxdomain no; # conform to RFC1035
   listen-on-v6 { any; };
   allow-recursion{mynetworks;};
   };
   [/sourcecode]
   ```

3. Restart the named daemon by issuing: /etc/init.d/bind9 restart

SD-B4 Setup

B4 Configuration



Network Interfaces

[sourcecode language="powershell"] # This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo eth1
iface lo inet loopback

iface eth1 inet static
address 192.168.1.1
netmask 255.255.255.0
[/sourcecode]

At this point, it is not necessary to configure eth0 (wan) since the configuration will be handled by the dhclientdaemon for both IPv4 and IPv6.

Configuration

The WAN interface in the B4 is configured by two separate instances of dhclient, one requesting an IPv6 address and the other an IPv4 address. However, the actions are chained by the execution of the first client, which calls a Client Relay Agent (CRA) and later a dhcpv4 client:

1. The first part of the configuration is done by a patched version of dhclient, which provides an option to override the default selection of the method to generate the dhcp unique identifier (DUID) (-D LL). The software is the same version that implements the TSV mode, so the same steps will be required, with the exception of the installation directory, which we will set at: /usr/local/sdb4

[sourcecode language="powershell"]

./configure –prefix=/usr/local/sdb4

[/sourcecode]

2. We now need to build and install the sdctl software, which implements an UPnP Internet Gateway Device that enables clients to request for port redirections. The steps required to set it up are:

[sourcecode language="powershell"] # cd rt28354/sdctl
./configure –prefix=/usr/local/sdb4
# make ; make install
[/sourcecode]

a. Copy the sample config file: rt28354/conf/sdb4-ubuntu/sdctld.conf into /etc/sdb4/confs

b. Edit the /etc/sdb4/confs/sdctld.conf file and replace the listening_ip directive with the correct IP address block of the local network:

[sourcecode language="powershell"] listening_ip=192.168.1.1/24
[/sourcecode]

3. Once the software is built and installed, we need to prepare the configuration so that the chain of commands are executed properly:

a. Create a directory to store the sdb4 configuration files and scripts: mkdir /etc/sdb4 ; mkdir /etc/sdb4/confs/; mkdir /etc/sdb4/scripts

b. Copy the sample configuration files and scripts into the newly created directories:

[sourcecode language="powershell"] # cp rt28354/conf/sdb4-ubuntu/dhclient6.conf /etc/sdb4/confs

```
# cp rt28354/conf/sdb4-ubuntu/dhclient4.conf /etc/sdb4/confs
# cp rt28354/conf/sdb4-ubuntu/dhclient6-script /etc/sdb4/scripts
# cp rt28354/conf/sdb4-ubuntu/dhclient4-script /etc/sdb4/scripts
# cp rt28354/conf/sdb4-ubuntu/setup6 /etc/sdb4/scripts
# cp rt28354/conf/sdb4-ubuntu/setup4 /etc/sdb4/scripts
```

[/sourcecode]

c. Edit the /etc/sdb4/conds/dhclient6.conf file and remove the reference to dnsmasq in the make_resolv_conf() function since it will not be used to serve dhcp requests.

d. Edit /etc/sdb4/script/setup4 remove lines 22 and 23 which are in charge of starting a local dhcp server for clients (not needed since using the distribution's), and edit line 39 and replace the reference to the bin directory for sbin.

[sourcecode language=”powershell” highlight=”22,23,39″] #!/bin/sh

```
# IPv4 part of the setup script, to be called with
# wan-interface-name lan-interface-name mapped-address
# port-range-min port-range-max

WANIF=$1
LANIF=$2
MAPPED=$3
PRMIN=$4
PRMAX=$5

echo "my WAN interface is:" $WANIF
echo "my LAN interface is:" $LANIF
echo "the global mapped address is:" $MAPPED
echo "the port range is": $PRMIN ".." $PRMAX

#set -x

sysctl -w net.ipv4.ip_forward=1
ip route add 224.0.0.0/4 dev $LANIF
touch /tmp/leases
/etc/sdb4/bin/dhcpd -q -4 -cf /etc/sdb4/confs/dhcpd4.conf -lf /tmp/leases

iptables -t nat -N SDCTLD
iptables -t nat -A PREROUTING -d $MAPPED -i tun0 -j SDCTLD
iptables -t filter -N SDCTLD
iptables -t filter -A FORWARD -i $LANIF ! -o $LANIF -j SDCTLD

iptables -t nat -A POSTROUTING -o tun0 -p tcp -j SNAT
–to-source $MAPPED:$PRMIN-$PRMAX
iptables -t nat -A POSTROUTING -o tun0 -p udp -j SNAT
–to-source $MAPPED:$PRMIN-$PRMAX
iptables -t nat -A POSTROUTING -o tun0 -p icmp -j SNAT
–to-source $MAPPED:$PRMIN-$PRMAX

export WANIF MAPPED SDPARAMS=direct/tcp:$PRMIN-$PRMAX/udp:$PRMIN-$PRMAX/

/etc/sdb4/sbin/sdctld -d -f /etc/sdb4/confs/sdctld.conf
```

[/sourcecode]

4. Edit the /etc/sdb4/scripts/setup6 file and change line 29 and 30 to reference to the sbin directory instead of bin:

[sourcecode language=”powershell” highlight=”29,30″] #!/bin/sh

```
# IPv6 part of the setup script, to be called with
# wan-interface-name my-address gateway-address aftr-end-point

# from
# /etc/sdb4/bin/dhclient -6 -q -nw -D LL -cf /etc/sdb4/confs/dhclient6.conf
# -lf /tmp/lease6 $WANIF

WANIF=$1
MYADDR6=$2
GW6=$3
AFTR6=$4

echo "my WAN interface is:" $WANIF
echo "my WAN IPv6 address is:" $MYADDR6
echo "the gateway address is:" $GW6
echo "the AFTR end-point address is:" $AFTR6

#set -x

ip -6 tunnel add tun0 mode ipip6 remote $AFTR6
local $MYADDR6 dev $WANIF encaplimit none
ip link set tun0 up
ip addr add 192.0.0.2 peer 192.0.0.1 dev tun0
ip route add default via 192.0.0.1
ip -6 route add default via $GW6

/etc/sdb4/sbin/dhccra -q -S node -i $WANIF $GW6
/etc/sdb4/sbin/dhclient -4 -q -nw -cf /etc/sdb4/confs/dhclient4.conf
-lf /tmp/lease4 $WANIF
[/sourcecode]
```

5. Since the current set of scripts, expect to find the executables in /etc/sdb4/bin and /etc/sdb4/sbin, create a symbolic link to those from the /usr/local/sdb4/bin and /etc/sdb4/sbin directories:

```
[sourcecode language=”powershell”] # ln –s /usr/local/sdb4/bin /etc/sdb4/bin
# ln –s /usr/local/sdb4/sbin /etc/sdb4/sbin
[/sourcecode]
```

DHCP server for Internal Clients

This server will be used to configure LAN clients. Since there is nothing special about this configuration, we can use the DHCP server available in the distribution. In this setup, we will use ISC’s DHCP server 4.1 available in the debian distribution.

The steps to configure the server are as follow:
1. Install the DHCP server software by issuing: aptitude install isc-dhcp-server
2. Edit /etc/default/isc-dhcp-server and specify eth1 in the interface list:

```
[sourcecode language=”powershell”] # Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#    Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
[/sourcecode]
```

3. Edit /etc/dhcp/dhcpd.conf to configure the subnet:

```
[sourcecode language=”powershell”] #
# Sample configuration file for ISC dhcpd for Debian
#
#

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages (‘none’, since DHCP v2 didn’t
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks…
option domain-name "example.org";
option domain-name-servers 149.20.74.100 ;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.100 192.168.1.200;
option routers 192.168.1.1;
}
[/sourcecode]
```

Note: the ip address 149.20.74.100 corresponds to the IP configured on the eth0 interface in the SD-AFTR box.

4. Start up the DHCP server by issuing the command: /etc/init.d/isc-dhcp-server start

5. At this point, the clients should be able to obtain an IP address by requesting it on the local network.

Running the Software

After all these steps have been followed, edit the /etc/rc.local file and add (before the exit instruction) the following command:

```
[sourcecode language="powershell"] /etc/sdb4/bin/dhclient -6 [-q -nw -D LL -cf /etc/sdb4/confs/dhclient6.conf -lf /tmp/lease6 eth0
[/sourcecode]
```

This will trigger the following actions:

1. Request an IPv6 address for the WAN interface and run the /etc/sdb4/scripts/dhclient6-script which configures the IP address in the interface and calls the /etc/sdb4/scripts/setup6 script.
2. The setup6 script then brings up the IPv4 over IPv6 tunnel and starts the DHCP Client Relay Agent (dhccra), and the dhclient in IPv4 mode, bound to the WAN interface. At this point the dhccra will encapsulate the DHCPv4 request into IPv6 packets, and sent them to the DHCP server running in TSV mode in the AFTR box.
3. At this point, the dhclient in IPv4 mode will receive a response from the DHCPv4 server in TSV mode with the SD parameters (IP address, port min-max, etc.), which will get passed to the /etc/sdb4/scripts/dhclient4-script for further configuration using the setup4 script.
4. The setup4 script finally configures netfilter (iptables) with the appropriate port ranges and starts sdctld which listens for client UPnP, PCP and NAT-PMP requests for port forwarding.

Table of Content

Requirement

AFTR Setup

SD-B4 Setup

OpenWrt SD-B4

Last modified: May 8, 2013 at 1:01 am

Contact

Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Email: info@isc.org
Phone: +1 650 423 1300
Fax: +1 650 423 1355

Tweets Follow

ISC
@ISCdotORG 4h

Walmart goes to war with Amazon over open source tek.io/1GWlNMm via @techrepublic #OpenSource ^TMJ

Show Summary

ISC
@ISCdotORG 22 Oct

Open source bugs could give hackers new attack tool cso.com.au/article/587272...
#opensource ^TMJ

Tweet to @ISCdotORG

Follow ISC

- Twitter
- Facebook
- LinkedIn
- YouTube