Linux as an IPv6 dual stack Firewall

Presented By: **Stuart Sheldon**
stu@actusa.net
http://www.actusa.net
http://www.stuartsheldon.org

IPv6

2001:0DB8:0000:0000:021C:C0FF:FEE2:888A

- Address format: Eight 16 bit hexadecimal groups separated by ':'s
- Total of 128 bits of address space available
- 2^128 or 340 Billion, Billion, Billion, Billion addresses
- Minimum network size /64 (18 Billion, Billion devices)
- Supports
  - Unicast Addresses
  - Multicast Addresses
  - Anycast Addresses

More IPv6

- Also Supports
  - Auto Client Configuration (Network Discovery)
  - Router Discovery / Advertising
  - Duplicate Address Detection
- Does Not Support
  - Network Broadcasts
  - Network Address Translation
  - Longer netmasks then /64
  - Packet Fragmentation

About Addresses

- Address Shortcuts
  - 2001:0DB8:0000:0000:0000:0000:0000:0001
  - Removing groups of '0' - 2001:0DB8::0001
  - Removing leading '0' - 2001:DB8::1

About Addresses

- Link Local Addresses
  - Every IPv6 interface must have one
  - Only used on local LAN.
  - Never routed
  - Multiple interfaces can have the same link-local address
  - When attaching to a link-local address, you must specify the interface you want to go out on

About Addresses

- Automatic Address Format (EUI-64)
  - <NetworkAddress> + <MAC-First-12>FFFE<MAC-Last-12>
  - Then Invert Bit 7 in the host portion of the address
- To specify an IPv6 address in a browser's address bar, you would enclose it in '[ ]' brackets. [2607:ff38:1::1b]

IPv6 Address Types

- Link-local unicast: FE80::/10
- Global unicast: 2000::/3
- Local IPv6 Addresses: FC00::/7
- Multicast: FF00::/8
- Loopback Address: ::1/128
- IPv4 Mapped: ::FFFF:192.168.1.100
- Router Anycast: <Global_Network>::
- Everything: ::/0

IPv6 Privacy

- RFC 4941 - Randomizes client IPv6 Global addresses to maintain client privacy.
  - On by default in Windows
  - Off by default in Linux
- Windows uses random addresses for auto configuration.

IPv6 Tunneling, Etc...

- Toredo – Automatic IPv6 Tunneling (2001::/32)
  - On by default in older Windows releases
  - Allows for global routing behind NAT (BAD)
- 6in4 Tunneling – Point-to-point IPv6 Tunneling.
  - Allows point-to-point tunneling of IPv6 data between network endpoints via IPv4
- 6to4 Tunneling – Network Tunneling (2002::/16)
  - Allows for auto tunneling between IPv6 networks through IPv4 networks (Limited Adoption)

Auto Configuration vs. DHCPv6

DHCPv6

- Pros
  - Address Tracking
  - Fixed Address Assignment
  - DNS Server Assignment
  - Dynamic PTR / AAAA Updates
- Cons
  - Complicated to implement
  - Client compatibility is mixed at best

Auto Configuration

- Pros
  - Setup is less complicated
  - Almost all clients supported out of the box
  - Less system overhead
- Cons
  - No Address Tracking

Address Daemon Packages

- DHCPv6
  - ISC DHCP-Server / Client
  - Wide DHCP-Server /Client
- Auto Configuration
  - Quagga
  - Router Advertisement Daemon (RaDvD)
  - RDNSsD (Client)

Our Target Setup

- Debian Squeeze GNU Linux
- 6in4 Tunnel from Tunnel Broker routing a /64
- Auto configuration using Quagga
- Firewall supplied by IPTables and IP6Tables

Hardware

Account Menu

Main Page
Account Info
Logout

User Functions

Create Regular Tunnel
Create BGP Tunnel
IPv6 Portscan

Create New Tunnel

You currently have 2 of 5 tunnels configured.

- If you are trying to reclaim a tunnel simply use your last IPv4 address here. If you have any issues please email ipv6@he.net.
- If you have a public ASN and wish to setup a full BGP feed, please use **this form** instead.

IPv4 Endpoint (Your side):

You are viewing from: 208.83.99.40

We recommend you use: Los Angeles, CA, US [66.220.18.42]

Available Tunnel Servers:

Asia	
○ Hong Kong, HK	216.218.221.6
○ Singapore, SG	216.218.221.42
○ Tokyo, JP	74.82.46.6
Europe	
○ Amsterdam, NL	216.66.84.46
○ Berlin, DE	216.66.86.114
○ Frankfurt, DE	216.66.80.30
○ London, UK	216.66.80.26
○ Paris, FR	216.66.84.42
○ Prague, CZ	216.66.86.122
○ Stockholm, SE	216.66.80.90
○ Warsaw, PL	216.66.80.162
○ Zurich, CH	Not Available (Full)
North America	
○ Ashburn, VA, US	216.66.22.2
○ Chicago, IL, US	209.51.181.2
○ Dallas, TX, US	216.218.224.42
○ Denver, CO, US	184.105.250.46
○ Fremont, CA, US	72.52.104.74
○ Fremont, CA, US	64.62.134.130
○ Kansas City, MO, US	216.66.77.230
◉ Los Angeles, CA, US	66.220.18.42
○ Miami, FL, US	209.51.161.58
○ New York, NY, US	209.51.161.14
○ Seattle, WA, US	216.218.226.238
○ Toronto, ON, CA	216.66.38.58

Create Tunnel

HURRICANE ELECTRIC
INTERNET SERVICES

Account Menu

Main Page
Account Info
Logout

User Functions

Create Regular Tunnel
Create BGP Tunnel
IPv6 Portscan

Tunnel Details

IPv6 Tunnel | Example Configurations | Advanced

Tunnel ID: 189431 — Delete Tunnel
Creation Date: Dec 31, 2012
Description: SCALE Talk Tunnel

IPv6 Tunnel Endpoints

Server IPv4 Address: 66.220.18.42
Server IPv6 Address: 2001:470:**c**:8bc::1/64
Client IPv4 Address: **208.83.99.40**
Client IPv6 Address: 2001:470:**c**:8bc::2/64

Available DNS Resolvers

Anycasted IPv6 Caching Nameserver: 2001:470:20::2
Anycasted IPv4 Caching Nameserver: 74.82.42.42

Routed IPv6 Prefixes

Routed /64: 2001:470:**d**:8bc::/64
Routed /48: **Assign /48**

rDNS Delegations — Edit

rDNS Delegated NS1:
rDNS Delegated NS2:
rDNS Delegated NS3:
rDNS Delegated NS4:
rDNS Delegated NS5:

Account Menu

Main Page
Account Info
Logout

User Functions

Create Regular Tunnel
Create BGP Tunnel
IPv6 Portscan

Tunnel Details

IPv6 Tunnel | **Example Configurations** | Advanced

Linux-route2

Copy and paste the following commands into a command window:

```
modprobe ipv6
ip tunnel add he-ipv6 mode sit remote 66.220.18.42 local 208.83.99.40 ttl 255
ip link set he-ipv6 up
ip addr add 2001:470:c:8bc::2/64 dev he-ipv6
ip route add ::/0 dev he-ipv6
ip -f inet6 addr
```

NOTE: When behind a firewall appliance that passes protocol 41, use the IPv4 address you get from your appliance's DHCP service instead of the IPv4 endpoint you provided to our broker.

The configurations provided are example configurations and may be different depending on the version of the OS or the tools you are using. If you have any issues getting your tunnel to work please contact us at **ipv6@he.net** and we will be happy to assist you.

What Will Be Modified

- Add IPv6 Tunnel to /etc/network/interfaces
- Add IPv6 Routed Network to /etc/network/interfaces
- Change net.ipv6.conf.all.forwarding to '1'
- Configure Quagga Daemon for auto configuration and change vtysh 'pager' settings

```
# The primary network interface
auto eth0
iface eth0 inet static
        address 208.83.99.40
        netmask 255.255.255.192
        gateway 208.83.99.1

auto eth1
iface eth1 inet static
        address 192.168.100.1
        netmask 255.255.255.0

iface eth1 inet6 static
        address 2001:470:d:8bc::1
        netmask 64

auto tb6in4
iface tb6in4 inet6 v4tunnel
        address 2001:470:c:8bc::2
        netmask 64
        local 208.83.99.40
        endpoint 66.220.18.42
        ttl 255
        up /sbin/ip -6 route add ::/0 via 2001:470:c:8bc::1 || true
```

Add Inside IPv6 Network

Add 6in4 Tunnel

```
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
```

Uncomment

Quagga Setup

touch /etc/quagga/zebra.conf

chown quagga: /etc/quagga/zebra.conf

echo 'export VTYSH_PAGER=more' >> /etc/bash.bashrc

vi /etc/quagga/daemons

```
# the daemon will not be started by /etc/init.d/quagga. The permissions should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be owned by
# group "quaggavty" and set to ug=rw,o= though. Check /etc/pam.d/quagga, too.
#
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

Change from no to yes

Quagga Setup

reboot

vtysh
config terminal

interface eth1
no ipv6 nd suppress-ra
ipv6 nd prefix 2001:470:d:8bc::/64
exit

write
exit

Warning Will Robinson!

- You now have a fully functional IPv6 gateway
- There is no firewall installed what so ever
- All devices on your network that can take advantage of IPv6 auto configuration are sitting on the open Internet!

OK! We have an IPv4 / IPv6 Router!
Now What?

Simple IPv4 Firewall Script

```
iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X
iptables -X -t nat
iptalbes -X -t mangle

iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT
iptables -A INPUT -i eth0 -p 41 -s 66.220.18.42/32 -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT

iptables -A INPUT -i eth0 -m state –state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -j DROP

# iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
iptables -A POSTROUTING -t nat -o eth0 -j SNAT –to-source 208.83.99.40

iptables -A FORWARD -i eth1 -j ACCEPT
iptables -A FORWARD -i eth0 -m state –state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -j DROP
```

Adding To Your Firewall

 # IPv4 Clear Rules

```
iptables -F
iptables -F -t nat
iptables -F -t mangle
iptables -X
iptables -X -t nat
iptalbes -X -t mangle
```

 # IPv6 Clear Rules

```
ip6tables -F
ip6tables -F -t mangle
ip6tables -X
ip6tables -X -t mangle
```

Adding To Your Firewall

 # Loopback and ICMP IPv4

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT
```

 # Loopback and ICMP IPv6

```
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A INPUT -p icmpv6 -i lo -j ACCEPT
ip6tables -A FORWARD -p icmpv6 -i lo -j ACCEPT
ip6tables -A INPUT -p icmpv6 -i eth1 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 -i eth1 -j ACCEPT
ip6tables -A INPUT -p icmpv6 -i tb6in4 -j ACCEPT
ip6tables -A FORWARD -p icmpv6 -i tb6in4 -j ACCEPT
```

Adding To Your Firewall

 # IPv4 Input Rules
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A INPUT -i eth0 -p 41 -s 66.220.18.42/32 -j ACCEPT
iptables -A INPUT -i eth0 -m state –state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -j DROP

 # IPv6 Input Rules
ip6tables -A INPUT -i eth1 -j ACCEPT
ip6tables -A INPUT -d ff01::/16 -j ACCEPT
ip6tables -A INPUT -d ff02::/16 -j ACCEPT
ip6tables -A INPUT -i tb6in4 -m state –state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -j DROP

Adding To Your Firewall

 # IPv4 Forwarding Rules
iptables -A FORWARD -i eth1 -j ACCEPT
iptables -A FORWARD -i eth0 -m state –state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -j DROP

 # IPv6 Forwarding Rules
ip6tables -A FORWARD -i eth1 -j ACCEPT
ip6tables -A FORWARD -i tb6in4 -m state –state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A FORWARD -j DROP

Running Public Servers

 # IPv4 Web Services

```
iptables -A PREROUTING -i eth0 -d 208.83.99.40/32 \
      -p tcp –dport 80 -j DNAT –to-address 192.168.100.100
iptables -A FORWARD -i eth0 -d 192.168.100.100/32 -p tcp –dport 80 -j ACCEPT
iptables -A PREROUTING -i eth0 -d 208.83.99.40 \
      -p tcp –dport 443 -j DNAT –to-address 192.168.100.100
iptables -A FORWARD -i eth0 -d 192.168.100.100 -p tcp –dport 443 -j ACCEPT
```

 # IPv6 Web Services

```
ip6tables -A FORWARD -i tb6in4 -d 2001:470:c:8bc::64/128 \
      -p tcp –dport 80 -j ACCEPT
ip6tables -A FORWARD -i tb6in4 -d 2001:470:c:8bc::64/128 \
      -p tcp –dport 443 -j ACCEPT
```

Questions???