IPv6 Primer

Gene Cronk – CISSP, NSA-IAM

SME – North American IPv6 Task Force

Systems Admin – The Robin Shepherd Group

gene@hacksonville.org

Why IPv6? Quick History

- **1992**
  - Internet Engineering Task Force (IETF)
    - Global shortage of IPv4 addresses
    - Technical limitations of IPv4
- **1993**
  - RFC1550 created
- **1995**
  - Next generation internet protocol (IPv6) chosen as IPng (IP Next Generation)

Why IPv6? Comparison with IPv4

- **IPv4**
  - 32 bit address space (4.3 billion *possible* addresses)
- **IPv6**
  - 128 bit address space (3.4 * 10^38 or 340 undecillion addresses)
    - ***64 billion* IPs for every *square centimeter* on earth**

Why IPv6? Comparison with IPv4

- **IPv4**
  - 20 some odd years ago
  - Many band-aids applied to address needs
- **IPv6**
  - Integrates many network improvements made over that time

Why IPv6? Comparison with IPv4

- **Stateless autoconfiguration**
  - **IPv4**
    - DHCP server is possible, but not mandatory.
  - **IPv6**
    - Automagic Link Local address as soon as you boot the machine (see RFC 2462)
    - Mechanism is similar to getting a 169.xxx.xxx.xxx address on boot in IPv4

Why IPv6? Comparison with IPv4

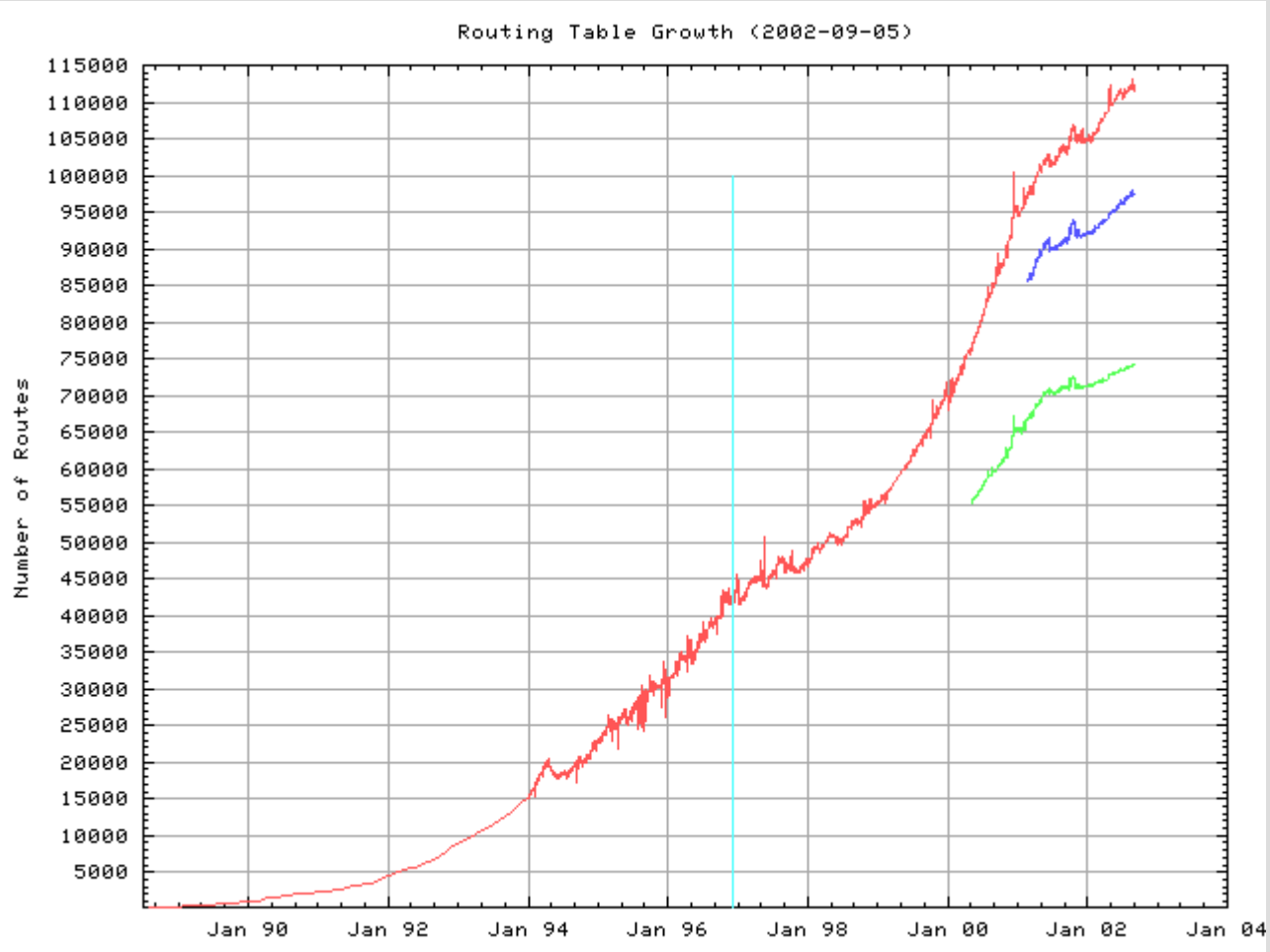
- **Security & QoS**
  - **IPv4**
    - IPSec and QoS are add-ons
  - **IPv6**
    - Encryption, IPSec and QoS built in

Why IPv6? Comparison with IPv4

- **A cure for routing table growth**
  - **IPv4**
    - Backbone routing table size has become a monsterous headache to ISPs and backbone operators. (113,000 as of 2003)
  - **IPv6**
    - Maximum amount of routes a router will see in the default-free zone is 8192.

Why IPv6? Comparison with IPv4

- From www.mcvax.org/~jhma/routing/bgp-hist.html :



Why IPv6? Comparison with IPv4

- **Roaming becomes much easier**
  - **Use of Mobile IPv6 and AnyCast**
    - Cell phone roaming much cleaner
    - Cell phone can automatically identify new routing information from a new tower
    - Cell phone keeps the same IP

Why IPv6? Comparison with IPv4

- **IPv6 reestablishes end to end connectivity**
  - The internet was originally designed for hosts to communicate directly with each other
  - One of the “fixes” to keep IPv4 working (NAT) breaks end to end connectivity

What do I need to know about IPv6?

- 6Bone
  - Experimental IPv6 beta network
  - IPv6 “Islands” connected via IPv4 tunnels
  - Connectivity
    - Native
    - Tunnel Broker and other tunneling methods
  - Number of networks continues to grow
    - http://www.cs-ipv6.lancs.ac.uk

What do I need to know about IPv6?

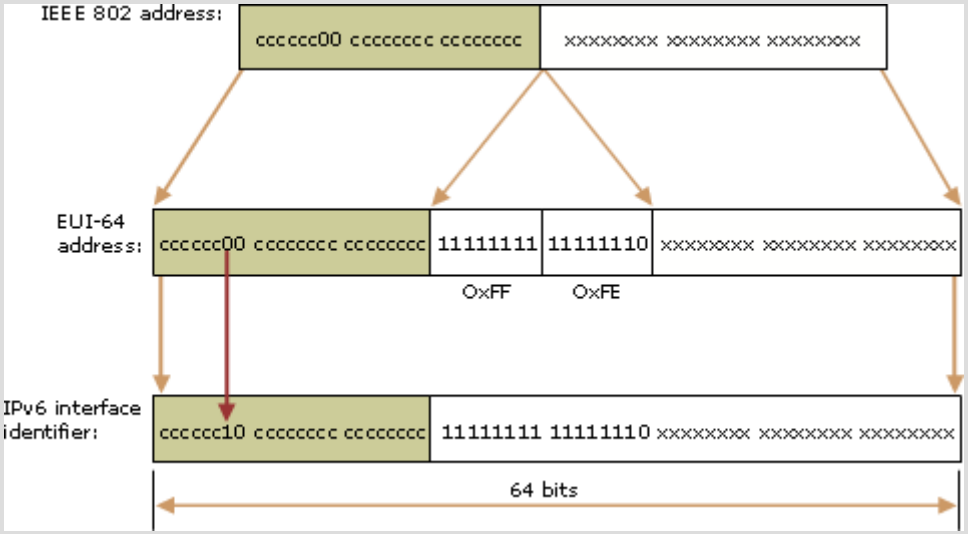
- Global adoption
  - Earliest adopters
    - Asia -- Japan & China expect full conversion by 2005
    - European Union
  - Resistive adopters
    - United States
      - US has roughly 70% of the world's IPv4 addresses
      - Detractors claim excessive level of effort to implement
  - New Developments
      - US DoD mandated all new network infrastructure equipment be IPv6 capable as of 10/2003
      - Full conversion for DoD expected by 2008
      - NTT/Verio, SpeakEasy, Hurricane Electric and others
      - Moonv6 Project (http://www.moonv6.org)

IPv6 – Addressing

- **3ffe:80ee:16f9:3481:efab:1092:aaaa:3ff1**
  - Each block in the address represents 16 bits
  - Just 2 words of an IPv6 address cover the entire IPv4 internet
- **The first word defines the type of address**
  - **3ffe** -- 6 Bone address (experimental globally routable IP)
    - Depreciated in leiu of 2001:: addresses (RFC 3701)
  - **fe80** -- Link Local address, used to get information about the network (routers, etc.)
  - **::1** -- localhost (127.0.0.1 in the IPv4 world)
  - :: -- equivalent to 0.0.0.0

IPv6 – Addressing
EUI-64

- Extended Unique Identifier (EUI-64)
- Clients can receive IPv6 address based on MAC
- 64 bit prefix assigned by Router Advertiser or DHCPv6, last 64 bits assigned by EUI-64
- FF-FE inserted between 3rd and 4th bytes
- 00-0B-3C-F4-22-CE becomes 00-0B-3C-FF-FE-F4-22-CE



IPv6 – Addressing
EUI-64

- Using MAC address as part of IP considered a privacy issue
- Not addressed in RFC 2373
- RFC 3041 describes a randomly-generated interface identifier that changes over time to provide a level of anonymity

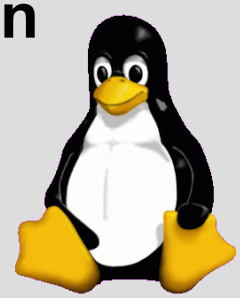
IPv6 – Addressing

- **2001** -- production globally routable IPv6 networks
- **2002** -- used for automatic 6to4 tunneling
- **FEC0** – (Site Local Address) equivalent to 192.168.xxx.xxx/24 or 10.xxx.xxx.xxx/8 addresses (DEPRECIATED).
    - To be replaced by FC00::/7
- **FF01, FF02 and FF05** are multicast addresses

IPv6 – OS Support

- **FreeBSD, OpenBSD, NetBSD, Apple OSX and BSDi -- include the Kame IPv6 stack**
  - http://www.kame.net
  - IPv6 is enabled by default

- **Linux kernel 2.4.xx -- very buggy IPv6 implementation**
  - can be augmented by patches from the USAGI project
  - http://www.linux-ipv6.org
  - USAGI is a port of the KAME project to Linux



- **Linux kernel 2.6.xx -- USAGI patches included by default**
- **Solaris 8.x and above -- native support**
- **Novell Netware 6.x and above – native support**
  - load BSDSOCK.NLM

IPv6 – OS Support

- **Windows 9x/Me** -- no Microsoft supported IPv6 capability
- **Windows NT 4** -- very early beta IPv6 stack
- **Windows 2000** -- beta quality IPv6 stack
- **Windows XP and Windows 2003 Server** -- IPv6 stacks built in
  - Typing **"ipv6 install"** in a command shell (Windows XP) or adding the stack in network properties (Windows XP/2003) enables these stacks.
  - **"netsh"** to control IPv6 from CLI (Windows XP/2003)

IPv6 – Tunnel Brokers

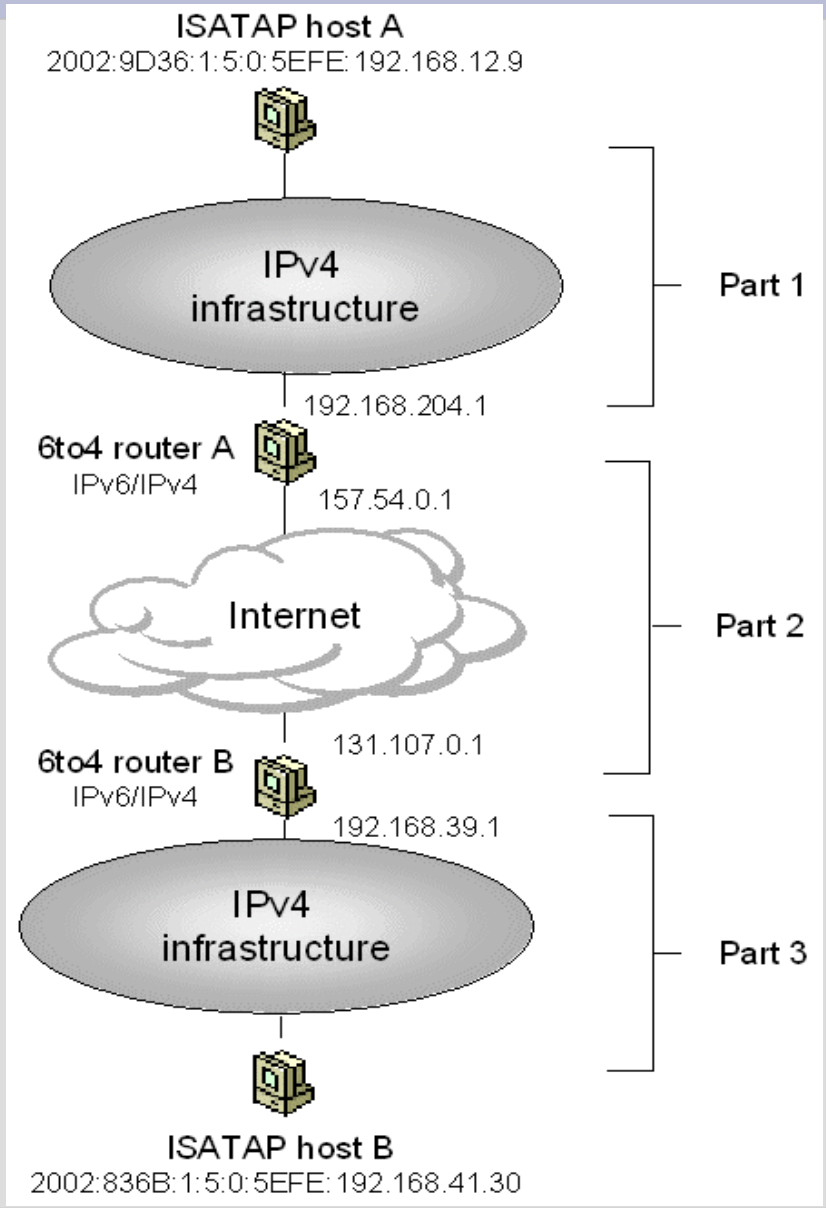
- **Top North American IPv6 Providers**
  - NTT/Verio, Freenet6, Hurricane Electric (SpeakEasy soon)
- **NTT/Verio**
  - Supplies tunnelling services to its customers in urban and rural areas
- **Hurricane Electric and Freenet6**
  - Open tunnelling servers
  - Anyone with an IPv4 address can tunnel IPv6
    - http://www.tunnelbroker.net (static v4 IP)
    - http://www.freenet6.net (dynamic v4 IP)
- **Other tunnel brokers available worldwide**
  - Most only require on-line registration

IPv6 – Tunnelling & Transition Methods

- **ISATAP** (Intra Site Automatic Tunnelling Addressing Protocol -- ::0:5EFE:w.x.y.z prefix)
- **6to4** (Tunnel Brokers – 3ffe or 2001 prefix)
- **Automatic 6to4** (Dynamic Tunnels – 2002 prefix)
- **Teredo and Silkroad** (6to4 tunnelling over UDP)
- **NAT/PT** (Network Address Translation/Protocol Translation)
- **Bump In The Stack/Bump In The API** (BIS/BIA)
- **Dual Stack Transitioning Mechanism** (DSTM)
- **Transport Relay Translator** (TRT)

IPv6 – Tunnelling & Transition Methods -- ISATAP

- Used mostly for IPv6 connectivty between hosts on a LAN, VLAN or WAN
- Requires a 6to4 gateway for packets to leave the local LAN
- Can be used for an IPv6 NAT
- IPv6 address includes IPv4 address.
  - 2002:836B:1:5:0:5EFE:10.40.1.29
- IPv4 becomes the link layer for IPv6



IPv6 – Tunnelling & Transition Methods -- 6to4 Via a Tunnel Broker

- Currently the most popular way to connect via IPv6
- Requires IP Protocol Type 41
- Does not work with NAT'ed IPv4 hosts unless the host is a 1 to 1 NAT
- Most tunnel brokers will give a /48 or a /64 subnet for the rest of your network
  - /48 = 1,208,925,819,614,629,174,706,176 IPs
  - /64 = 18,446,744,073,709,551,616 IPs
- Very easy to set up and change
- Frequently used as an attack vector, since tunnels can be set up to different countries easily
- http://www.sixxs.net has a 6to4 proxy that only shows the IPv6 source address

IPv6 – Tunnelling & Transition Methods -- 6to4 Via Auto Tunnelling

- Fairly easy to set up
- Convert your IPv4 address to hex, then put a 2002 in front of it:
  - 69.3.46.44 becomes 2002:4503:2e2c::/48
  - One IPv4 IP becomes a /48 subnet for IPv6
- Set default route for IPv6 traffic to 192.88.99.1
- Uses BGP to find the nearest 6to4 router and connect to the IPv6 internet
- Security questionable
  - you have little choice where your traffic is routed
- Windows XP SP1 auto tunnels by default
- Not included with OpenBSD

IPv6 – Tunnelling & Transition Methods -- Teredo

- Allows IPv6 tunnelling though NAT servers using UDP traffic
  - Uses port 3544 UDP by default
  - Port can be changed (to say, ports 53 or 500?)
- Microsoft and FreeBSD have the only implementations.
  - Windows XP SP1 only has a Teredo client
  - Kernel modules have been written for FreeBSD to be both a server and relay
- Could very easily be used as an attack vector
  - UDP traffic not commonly monitored
  - UDP not locked out at firewalls
- Considered a “last ditch” IPv6 tunnelling mechanism
- Draft calls for use of 3FFE:831F::/32 ONLY
- Does not allow tunnelling through restricted NATs

IPv6 – Tunnelling & Transition Methods -- SilkRoad

- Allows for IPv6 tunnelling though NAT servers using UDP traffic
  - Uses port 5188 UDP currently
- No current implementations
- Could also easily be used as an attack vector
- Allows for any address range to be used (not just 3ffe:831F::/32)
- VERY new draft
- Allows tunnelling through any type of NAT server

IPv6 – Tunnelling & Transition Methods -- NAT/PT

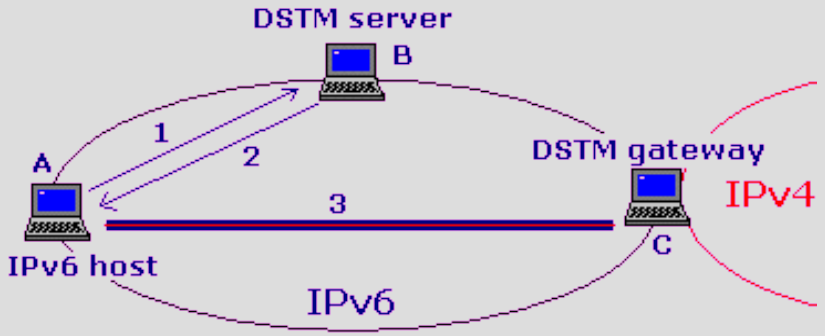
- **Network Address Translation/Protocol Translation**
- RFC 2766
- IPv6 hosts send requests to a dual-stacked gateway
- Gateway decides if the remote address is IPv4 or IPv6
  - Routes the packet as normal if destination is IPv6
  - Converts the packet to IPv4 with special header information if the destination is IPv4
  - Converts the returning packet to IPv6 and routes it back to the originating host
- Cisco has the only production quality implementation
- Technology is similar to an IPX/SPX only network connecting to hosts on the IPv4 internet

IPv6 – Tunnelling & Transition Methods -- BIS/BIA

- **Bump in the Stack/Bump in the API**
- Bump in the Stack – RFC 2767
- Bump in the API – RFC 3338
- Used on dual stacked hosts to proxy programs that are IPv4 only or IPv6 only to use the other protocol
- Security is questionable
- Windows XP and 2003 include “port proxy”

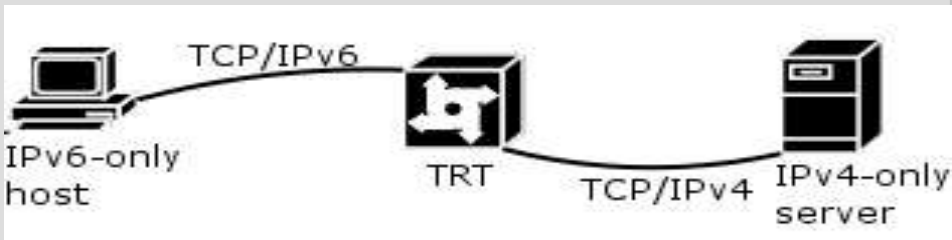
IPv6 – Tunnelling & Transition Methods -- DSTM

- **Dual Stack Transitioning Mechanism**
- Based on dynamic IPv4 over IPv6 tunnels.
- Temporary assignment of global IPv4 addresses to IPv6 hosts.
- Allows IPv4 only apps to run in an IPv6 environment
- Requires DSTM gateway and server
- Multi-platform
- Minimizes need for IPv4 IPs



IPv6 – Tunnelling & Transition Methods -- TRT

- **Transport Relay Translator**
- Works as a DNS Proxy
- TRT server translates IPv4 addresses to IPv6
  - fec0:0:0:0:ffff::/64 + 193.99.144.71 becomes
  - fec0:0:0:0:ffff:0:0:c163:9047
- BSD and Linux implementations
- Based on TOTD and FAITHD (BSD) or pTRTd (Linux)
- RFC 3142



IPv6 – Router Advertising

- **Allows your IPv6 border router to broadcast its existence**
  - Advertises to clients their IPv6 prefix and default route
- **Differs from DHCP**
  - Can only broadcast a default route and address prefixes
  - Cannot assign DNS, WINS, etc.
- **RA server is available in most IPv6 capable OSes**

IPv6 – DHCPv6

- Combines the functionality of router advertising and DHCPv4
- Currently in alpha stages in most implementations
  - Cisco's DHCPv6 stack -- considered production quality
- Provides prefix delegation
- Facilitates distribution of IPs, default routes, DNS and WINS servers, and other options available in DHCPv4

IPv6 – Security

- Firewalling IPv6
  - IPFW, PF for *BSD
  - ip6tables for Linux
  - Built-in firewall in Windows XP
    - Controlled by the “**netsh**” command
    - No IPv6 firewall in Windows 2003

IPv6 – Security

- Windows 2003
  - No IPv6 support
- Windows XP before Advanced Networking Pack
  - No IPv6 support in firewall
- Consumer Firewall Applications for Windows
  - Most HIDS to date
    - MIGHT pick up 6to4 traffic or IPv4 DNS lookups
    - Do not defend against native IPv6 traffic

IPv6 – Security

- Blocking UDP and IP Protocol Type 41 traffic
- Scanning for router advertisements using tools such as Ethereal
- As always, if not using the protocol, don't enable it...

IPv6 – DNS

IPv4 – A Record

www.hacksonville.org A 192.168.254.111

IPv6 – AAAA Record

www.hacksonville.org AAAA \
FEC0:0010:0083:1211:0000:0000:1287:123F

IPv6 – Applications

- IPv4 only applications can sometimes be patched and recompiled with IPv6 support
- IPv4 applications can be proxied to use IPv6 addresses
- Apps must be able to handle “:” in addresses
- Apps should be able to handle both IPv4 and IPv6 addresses

IPv6 – Sample Code

IPv4 only

- Uses IPv4 specific libraries and system calls

```
int i, s;
struct hostent *hp;
struct servent *sp;
struct sockaddr_in sin;
s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
hp = gethostbyname("www.hacksonville.org");
sp = getservbyname("http", "tcp");
for (i = 0; hp->h_addr_list[i]; i++) {
   memset(&sin, 0, sizeof(sin));
   sin.sin_family = AF_INET;
   sin.sin_len = sizeof(sin);
   sin.sin_port = htons(sp->s_port);
   memcpy(&sin.sin_addr, hp->h_addr_list[i], hp-
   >h_length);
   if (connect(s, &sin, sizeof(sin)) < 0)
      continue;
   break;
}
```

IPv6 – Sample Code

Dual Stack – IPv4 & IPv6

- Code uses DNS server to determine the IP address

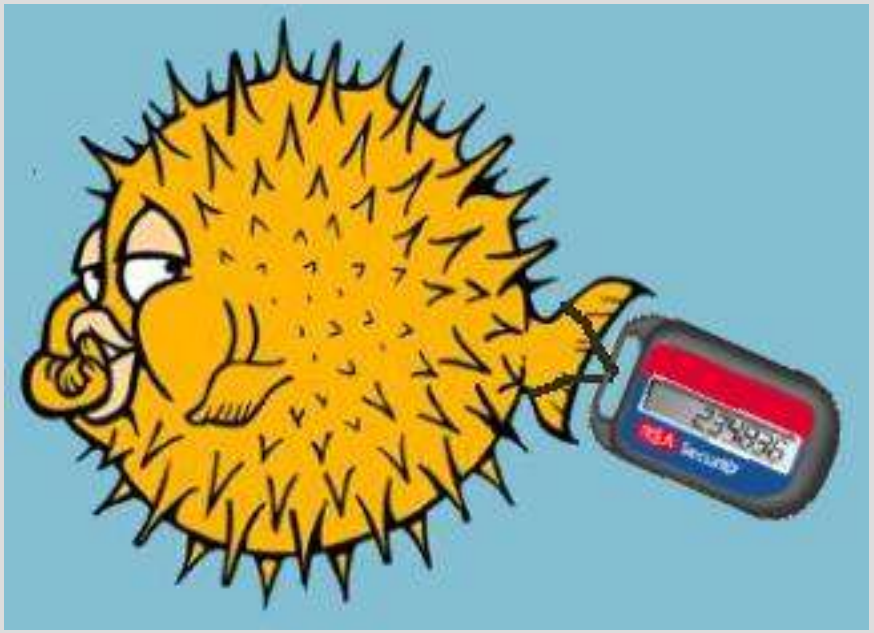
```
int s;
struct addrinfo hints, *res, *res0;
memset(&hints, 0, sizeof(hints));
hints.ai_family = PF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
getaddrinfo("www.hacksonville.org", "http", &hints,
   &res0);
for (res = res0; res; res = res->ai_next) {
   s = socket(res->ai_family, res->ai_socktype, res-
   >ai_protocol);
   if (connect(s, res->ai_addr, res->ai_addrlen) < 0) {
      close(s);
      continue;
   }
   break;
}
freeaddrinfo(res0);
```

IPv6 – Running Services (Apache)

- Requires service be IPv6 capable and listening
- Apache 2.xx, httpd.conf:
  - Listen 0.0.0.0:80
  - Listen [::]:80

IPv6 – Running Services (SSHD)

- SSHD, sshd_conf:
  - Port 22
  - Protocol 2
  - ListenAddress 0.0.0.0
  - ListenAddress ::



IPv6 – 12 Steps for Overcoming NAT Addiction

- **1.** We admit we are powerless over NAT - that our IP networks have become unmanagable.
- **2.** We come to believe that a power greater than NAT could restore us to security.
- **3.** Made a decision to turn our will and our networks over to the care of IPv6 as we understand it.
- **4.** Made a searching and fearless penetration test of our networks.
- **5.** Admitted to the CIO, to ourselves, and to another systems administrator the exact nature of our network security issues.
- **6.** Were entirely ready to have IPv6 remove all the defects of our IPv4 NAT'ted networks.

IPv6 – 12 Steps for Overcoming NAT Addiction

- **7.** Humbly asked router advertisements to remove our NAT shortcomings.
- **8.** Made a list of all networks we had harmed, and became willing to install IPv6 stacks on them all.
- **9.** Restored end to end connectivity to such networks whenever possible.
- **10.** Continued to take a network inventory and when we were using NAT promptly admitted it.
- **11.** Sought through network scans and DHCPv6 to improve our network connectivity with IPv6 as we understood it, Googling only for knowledge of IPv6 for us and the power to carry that out.
- **12.** Having had a router awakening as the result of these steps, we tried to carry this message to NAT addicts, and to practice these principles.

IPv6 – Links

North American IPv6 Task Force -- www.nav6tf.org
Linux IPv6 HowTo -- www.bieringer.de/Linux/IPv6
FreeNet6 Tunnel Broker -- www.freenet6.net
Hurricane Electric Tunnel Broker -- www.tunnelbroker.net
NetBSD -- www.netbsd.org/Documentation/network/ipv6
FreeBSD -- www.freebsd.org
OpenBSD -- www.openbsd.org
Kame (*BSD IPv6 Project) -- www.kame.net
USAGI (Linux Port of Kame) -- www.linux-ipv6.org
Japanese IPv6 Info Site -- www.ipv6style.jp/en/index.shtml
Windows IPv6 Ports -- win6.jp
IPv6 Events and News -- www.ipv6forum.com
Moonv6 Project -- www.moonv6.org

IPv6 – Questions?

Comments?

Concerns?