Introduction to IPv6 Firewalls With Linux and iptables

By Timothy D. Morgan

About Me

- I'm a long-time programmer, systems administrator, and security consultant who:
  - has dabbled in IPv6
  - uses Linux and free software as much as possible
  - is not a networking god, so don't ask me about routing protocols.
- I currently work for a security consulting company (www.vsecurity.com). Opinions you see here are my own.
- You may reach me at:

  tim-pdxlug (α) sentinelchicken.org

Motivation

- Based on current trends, it is estimated that all unallocated IPv4 address space will be consumed by November 29, 2011 [v4rep].
- Internet Protocol version 6 (IPv6) is the next generation Internet protocol designed to replace the current IPv4 protocol.
- IPv6 supports a 128-bit address space which is large enough to support an enormous amount of Internet expansion. It also features a number of minor improvements designed to streamline Internet communications.

IPv4 Address Allocation

- The current allocation algorithm is roughly:
  - IANA has a pool of reserved addresses which are distributed to the five Regional Internet Registires (RIRs).
  - RIRs request additional address space when they reach >80% utilization. They are then given an additional /8 block.
  - RIRs distribute address blocks ("prefixes") to Local Internet Registries (LIRs) and large ISPs based on their own policies.
- Based on this model, IANA will run out of address space first, which will then trickle down to the RIRs and LIRs.

Address Exhaustion

- Current estimates (as of April 25, 2008) indicate that IANA will run out of unallocated address space on January 13, 2011.
- Shortly thereafter, around November 29, 2011, the various RIRs will begin running out of address space.
- At that point, does the sky fall? Not really, because:
  - Many organizations have far more address space than they need.
  - NAT/NAPT will likely become even more popular.

Address Exhaustion (cont.)

- How accurate are the exhaustion estimates?
  - On one hand, there remains a large amount of inefficiently allocated address space. This could be reclaimed over time and allocated more fairly.
  - On the other hand, as we near the exhaustion, a "bank run" of sorts on the address space may ensue, which could accelerate allocation.
- What about an IP address market?
  - Unfortunately, IP addresses aren't completely portable. This could increase route fragmentation.
  - You think domain name "squatting" is bad? Just wait until we run low on addresses in an open market...

IPv6 Features

- Much larger address space
- No need for NAT/NAPT
- Streamlined, variable-length header
- Supports integrated IPSEC headers
- Better load balancing through anycast

IPv6 Features (cont.)

- Address autoconfiguration
- More flexible central management with DHCPv6
- Mobile device support
- Integration of various routing features:
  - QoS flow labels
  - PMTU is required for fragmentation

IPv6 Addresses

- Basic addresses consist of eight 16-bit blocks written in hexadecimal and delimited by ':'. For example:

  0123:4567:89AB:CDEF:0123:4567:89AB:CDEF

- "That's an awful lot to type", I hear you say. Well, there are some shortcuts:
  - In each 16-bit block, leading 0's can be omitted.
  - A variable length string of 0 blocks can be abbreviated as "::". (Limit one per address.)

- For example:

  2001:0001:0002:0000:0000:0000:0123:4567 becomes 2001:1:2::123:4567

IPv6 Addresses (cont.)

- Addresses may sometimes need to be contained in brackets for disambiguation. For instance:

  http://**[2002:C0A8:101::]**:80/some/path/

- Some special addresses:

Type	**Range**	**Analogous To**
Loopback	::1	127.0.0.1/8
Global Unicast (currently)	2000::/3	current unicast
Unique Local Unicast	FC00::/7	RFC 1918 ranges
Link Local Unicast	FE80::/10	N/A
Multicast	FF00::/8	224.0.0.0/4

v4/v6 Header Comparison

IPv4					
Bits	**0-3**	**4-7**	**8-15**	**16-18**	**19-31**
0	Version	Header Length	ToS	Length	
32	Fragment ID			Flags	Frag. Offset
64	TTL		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				
160	Options (Between 0 and 320 bits of option records, in 32-bit chunks)				
192					
224					
256					
288					
320					

IPv6							
0-3	**4-7**	**8-11**	**12-15**	**16-19**	**20-23**	**24-27**	**28-31**
Version	Traffic Class		Flow Label				
Payload Length				Next Header		Hop Limit	
Source Address							
Destination Address							
Next Header							

DNS Support For IPv6

- In order to support IPv6 addresses, a basic address record, "AAAA" was added and is analogous to IPv4's "A" records.
- Reverse DNS information (PTR records) is stored under the ".ip6.int" name space. Addresses are notated in reverse order (as with IPv4), one hexadecimal character at a time.
- Also, "A6" and "DNAME" records have been proposed, but much controversy has ensued over the wisdom of these record types, and their status is unclear.

Autoconfiguration and DHCPv6

- ICMPv6 provides two types of autoconfiguration.
  - Stateless: Routers to advertise a subnet prefix. Hosts use this prefix along with local information (MAC addresses) to generate their own address.
  - Stateful: Much like traditional DHCP under IPv4.
- DHCPv6 is updated with additional features:
  - Servers may use link-local or site-local multicast addresses.
  - Servers may push out updates without waiting for clients to refresh leases.
  - Authentication support.

Routing Protocol Changes

- ICMPv6 is a rewrite of ICMP. Changes include:
  - Integrated neighbor discovery, eliminating the need for ARP.
  - Integrated IGMP management.
  - ICMPv6 is considered a part of IPv6 and must be fully implemented by all nodes.
- RIPng, OSPF version 3, and BGP-4 w/ IPv6 extensions all exist to support IPv6 routing.
- The lack of fully private networks (RFC 1918 ranges) removes some pitfalls, such as route leakage, from route management.

Upper Layer Protocol Changes

- Some popular TCP & UDP protocols embed address information in their communications. This cross-layer behavior means these protocols will need to be updated. Examples include:
  - FTP
  - NIS/NIS+
  - IRC (w/ DCC)
  - NetBIOS (i.e. SMB on ports 137,138,139)
  - SOCKS
  - BitTorrent

Transition Technologies

- IPv6 designers envision a lengthy transition process which comes in multiple steps.

- It is likely that most systems will be "dual-stack", meaning they will support both IPv6 and IPv4 for the forseeable future.

- Multiple tunneling technologies are available to allow users and systems administrators choice and flexibility in communicating with the IPv6 Internet during the transition.

Static Tunnels

- Until your ISP offers you direct routes to the IPv6 Internet, how can you communicate to IPv6 hosts?
- The most obvious answer is to set up a static tunnel. Given a peer who is willing to route your IPv6 packets, simply send them IPv4 packets with embedded IPv6 packets.
- There are several free IPv6 tunnel brokers around the world. Users/administrators simply need to sign up for an account and then configure routers to use the tunnel.

6to4 Automatic Tunneling

- Static tunnels are a bit of work to set up and maintain with a tunnel broker.
- Also, what if your nearest tunnel broker today is 3000 miles away, but next week another broker comes online right next door? This can be a pain to manage.
- 6to4 is an automatic tunneling mechanism which is easier to set up because it does not require a tunnel broker account.
- Your 6to4 connectivity should gradually improve over time because it does not require you to use a specific tunnel broker.

6to4 Addressing

- Anyone with an IPv4 address is automatically assigned a 6to4 address range in 2002::/16.
- Your 6to4 block is determined by your IPv4 address. If you had an address of "222.173.190.239", your 6to4 prefix would be:

  2002:**DEAD:BEEF**::/48

- Generally 6to4 tunneling needs to be set up at the border of networks, particularly if IPv4 NA(P)T is used.

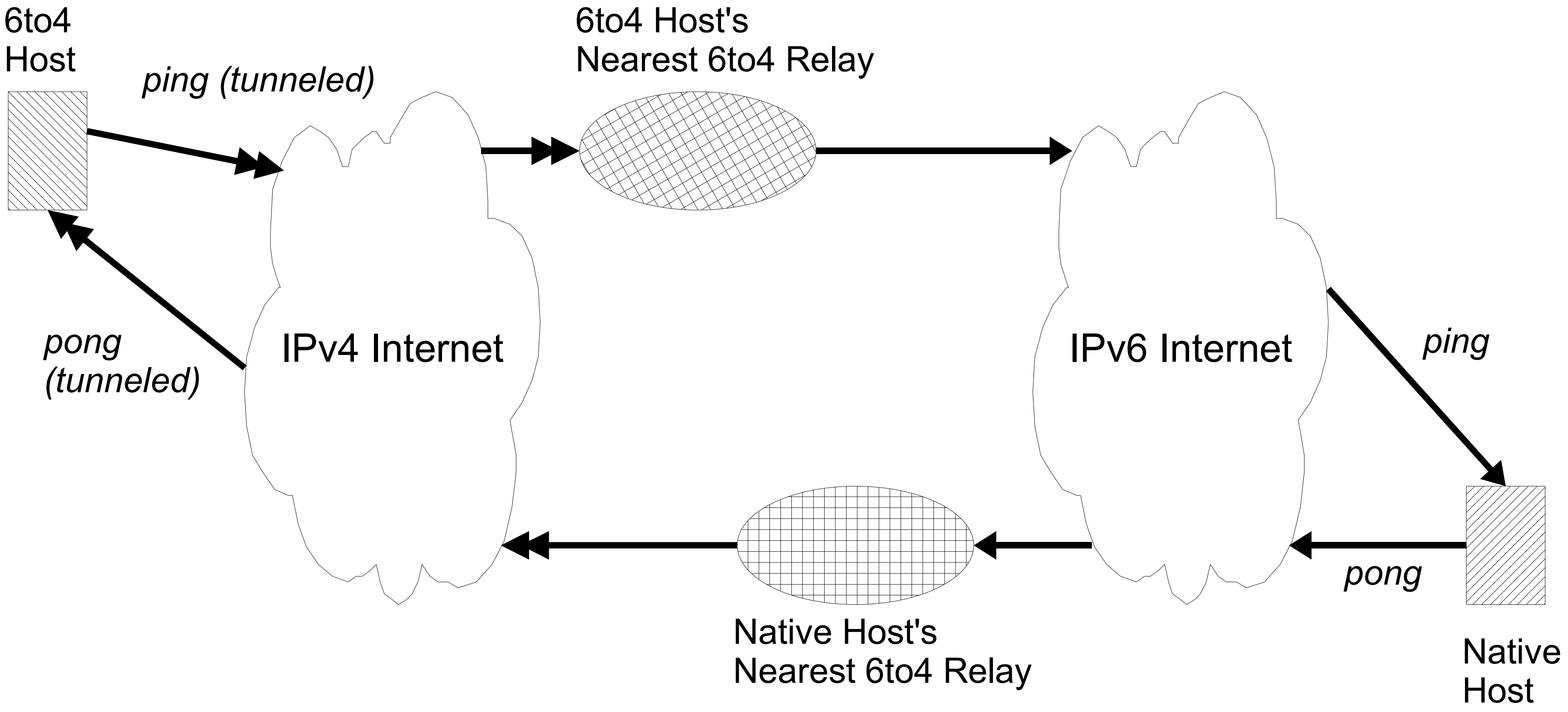
6to4 to 6to4 Routing

- As packets leave a network, the 6to4 gateway wraps them in IPv4 headers. How is this wrapper constructed?
  - The IPv4 source address can be easily pulled from the IPv6 source address.
  - If the destination IPv6 address is also a 6to4 address, then the destination IPv4 address is similarly extracted.
- This direct routing over the IPv4 cloud allows for reliable, distributed tunnels with relatively little overhead.

6to4/Native & Native/6to4 Routing

- In order to communicate with the native IPv6 Internet, we must rely on some relay routers.
- 6to4 uses specially reserved IPv4 and IPv6 anycast addresses for this purpose.
  - 6to4 sourced packets heading to non-6to4 addresses are sent to the IPv4 anycast address.
  - Native sourced packets heading to 6to4 addresses are sent to the IPv6 anycast address.
- This kind of routing ends up being inherently asymmetric, which may be slower and difficult to debug.

6to4 Asymmetric Routing



Teredo

- What if IPv4 firewalls block all embedded protocols except UDP/TDP/ICMP by default? What if a user is behind NAPT and can't receive 6to4 packets?
- Teredo is a Microsoft-backed tunneling mechanism designed to help end-users get around firewalling issues while requiring very little knowledge of networking [tdo1].
- Teredo accomplishes this by embedding IPv6 packets in UDP. It also uses a number of NAT-traversing techniques to improve the likelihood of connectivity on diverse networks.

Teredo (cont.)

- Windows Vista has Teredo enabled by default. Vista tries very hard to obtain an IPv6 address and to use it for communications.
- While Teredo is highly convenient and will likely be very popular early during the transition, it also has a lot of overhead.
- Many network administrators have expressed concern over Teredo, because it can easily open up unfettered direct connectivity to their organizations' desktop systems over IPv6.
- The Miredo project implements Teredo under Linux and BSD platforms [mir].

NAPT-PT

- Let's review some definitions:
  - NAT – Network Address Translation
    - IP addresses translated
  - NAPT – Network Address and Port Translation
    - IP addresses and ports translated as needed. Also commonly referred to as simply "NAT", but this is a misnomer.
- NAPT-PT is just NAPT with protocol translation (between v4 and v6) added in.
  - In theory, this could be used to help legacy IPv4-only hosts communicate with IPv6 hosts, and vice versa.
  - In reality, it probably won't be used much.

Other Transition Technologies

- A number of other tunneling systems have been proposed, but most have not got off the ground. Some of these include ISATAP and RFC2893 "automatic" tunnels. Others may be introduced to address specific problems.
- For some ISPs, it may be feasible to simply proxy specific services, and limit others to IPv6 only.
- As the IPv4 Internet wanes, we will likely start to see IPv4 tunneled in IPv6. Fortunately the groundwork for this has been laid.

Remaining Deployment Challenges

- While many DNS TLDs have added support for AAAA record glue, many others have not. The following are known to support AAAA glue:
  - .com, .net, .biz, .info, .mobi, .name, .travel, .au, .be, .ch, .cl, .cn, .de, .eu, .fr, .hu, .ie, .li, .lt, .nl, .nz, .se, .tw, .uk.
  - Even with this TLD level support, it is difficult to find a registrar who supports AAAA glue records.
- Client software:
  - Large volumes of software still don't support IPv6.
  - Software that supports both IPv4 and IPv6 often fails to drop back to IPv4 gracefully.

Challenges: Breaking the Nash Equilibrium

- An inefficient equilibrium exists between content providers and users:
  - Users have no incentive to adopt because there is no benefit in doing so with no content available.
  - Content providers have no incentive to adopt because there are no users using it.

		Users	
	Action	**None**	**Deploy IPv6**
Content Providers	**None**	Users: moderate payoff, little investment Content Providers: moderate payoff, little investment	Users: moderate payoff, moderate investment Content Providers: moderate payoff, little investment
	Deploy IPv6	Users: moderate payoff, little investment Content Providers: moderate payoff, moderate investment	Users: high payoff, moderate investment Content Providers: high payoff, moderate investment

Challenges: Reducing Costs

- "The problem is so big, what can I possibly do?"
  - Learn about IPv6.
    - As the number of IPv6-savvy people grows, the cost for businesses to support it should drop.
    - There may be another Y2K-like employment blitz. Being the only geek on the block with IPv6 knowledge could net you a bigger paycheck. ;-)
  - Forcing users to deal with tunneling is a pain. Ask your ISP about IPv6 support.
    - They'll probably laugh, if you're lucky. Otherwise they'll ask you: "What is IPv6?".
    - If enough of us ask, maybe they'll start considering pilot projects.

Challenges: Increasing Benefits

- Content providers need statistics telling them users are using it, so be a user. Find as many IPv6 web sites as you can and use them. Some interesting ones:
  - ipv6.google.com, ipv6experiment.com
- Better yet, be an IPv6 content provider. If you already host web sites, setup IPv6 versions.
  - Currently, many IPv4 users can be negatively impacted if you simply add an AAAA record with the same host name.
  - Instead, most sites use an ipv6.* prefix and link to it from the IPv4 site.

IPv6 Firewalling with Linux

- Assumptions
  - We're working with a basic IPv4 NAPT network with one external IPv4 address.
  - No network redundancy or routing protocols. Typical small business or home networking setup.
  - Using 6to4 or static tunnel for connectivity.
- Prerequisites
  - Kernel support:
    - Version 2.6.21+, or recent netfilter patches
    - IPv6 support enabled
    - All ip6tables modules compiled
  - iptables package installed

Configuring a Static Tunnel

- Once you register for an account with a tunnel broker, you'll typically be asked to provide your IPv4 gateway address and will be assigned:
  - an IPv4 tunnel IP
  - IPv6 tunnel network (often a /64 range) and end point addresses
  - an IPv6 block to use for your internal hosts (a /64 or /48 range
- Some brokers may require you to use special software for managing tunnels. In that case, follow their instructions for configuration. See [fn6,sixxs1,tbf1] for more information.

Configuring a Static Tunnel (cont.)

- Once you have this network information, configure your Linux router/firewall with:

```
ip tunnel add v6tun mode sit remote V4REMOTE
  local V4LOCAL ttl 255

ip link set v6tun up

ip addr add V6LOCAL dev v6tun

ip route add ::/0 dev v6tun

ip addr add V6RANGE dev eth0
```

Configuring 6to4

- 6to4 tunnels have a very similar configuration, but require no registration:

```
ip tunnel add tun6to4 mode sit remote any
  local V4LOCAL ttl 255

ip link set dev tun6to4 up

ip -6 addr add 2002:V4HEX::/128 dev tun6to4

ip -6 route add 2000::/3 via ::192.88.99.1 dev
  tun6to4 metric 1
```

Configuring 6to4 (cont.)

- On Debian-based systems, one can setup `/etc/network/interfaces` to start up a 6to4 tunnel at boot:

```
auto tun6to4
iface tun6to4 inet6 static
 pre-up /sbin/ip tunnel add tun6to4 mode sit
  ttl 255 remote any local V4LOCAL
 address 2002:V4HEX::
 netmask 16
 gateway 2002:c058:6301::1
 post-up /sbin/ip -6 route add 2000::/3
  via ::192.88.99.1 dev tun6to4 metric 1
```

Tim's Firewall

- Consists of a collection of shell scripts organized by protocol and table:

```
start.sh
ipv4/start.sh
ipv4/filter.sh
ipv4/nat.sh
ipv6/start.sh
ipv6/filter.sh
```

- Basic networking settings, interface configurations, and protocol-specific settings are stored in the various `start.sh` scripts.
- Table-specific rules are in `filter.sh` & `nat.sh`

Tim's Firewall (cont.)

- Bulk of important rules stored in `filter.sh`.
  - All packets first run through a demultiplexing stage to help organize the rulesets and to prevent spoofing. The pseudocode for this looks like:

```
If packet came from outside and the source address is a
  reserved range, drop.

If packet came from network A on network A's interface and
  is headed to network B, jump to "A-B" chain.

If packet came from network B on network B's interface and
  is headed to network A, jump to "B-A" chain.

If packet came from network C on network C's interface and
  is headed to network A, jump to "C-A" chain.

...

Otherwise, drop since it must be spoofed.
```

- This results in N*(N-1) chains for N networks.

Tim's Firewall (cont.)

- A typical pair of unidirectional chains look like:

```
$IPTABLES -N trusted-outside
$IPTABLES -A trusted-outside -j ACCEPT


$IPTABLES -N outside-trusted
$IPTABLES -A outside-trusted -m state --state
  INVALID -j logdrop
$IPTABLES -A outside-trusted -m state --state
  ESTABLISHED -j ACCEPT
$IPTABLES -A outside-trusted -p icmp -j ACCEPT
$IPTABLES -A outside-trusted -j logdrop
```

Changes to IPv4 Firewall

- In order to ensure we receive IPv6 packets which are embedded in IPv4 packets, we should add an explicit rule to accept these:

```
$IPTABLES -A INPUT -i $OUTSIDE_IF -p 41 -j
  ACCEPT
```

- Note that this should only allow tunneled packets to reach the border, not be forwarded to the inside network.
- Depending on your policy for Teredo use, you may want to explicitly allow or restrict traffic directed at port 3544/UDP externally.

IPv6 Debugging Tools & Resources

- `ping6`, `telnet6`, `traceroute6`, and `tracepath6`
  - Generally operate the same as their v4 counterparts
- Most DNS query tools (`host`, `dig`, `nslookup`, etc) support v6 record types, though refer to documentation to see if v6 transport is supported for lookups.
- IPv6 routes can be checked/updated with:
  - "`route -6 ...`" or "`ip -6 route ...`"
- `tcpdump` is very helpful for debugging tunnels on each side of a border device since it can parse IPv6 packets when tunneled.

Tools & Resources (cont.)

- "Looking Glass" services allow one to debug routes from multiple locations via simple web forms or publicly accessible routers. Two popular IPv6 services are [helg] and [sixxs2].
- If all else fails, try posing questions:
  - on IRC at freenode.net/#ipv6
  - to the cluenet.de ipv6-ops mailing list [v6ops].

Questions?

- Slides and other materials are available at:

  http://projects.sentinelchicken.org/howtos/lug-ipv6/

References

fn6 What is Freenet6?
http://go6.net/4105/freenet.asp

helg Hurricane Electric – Looking Glass
http://lg.he.net/cgi-bin/index.cgi

mir Miredo: Teredo for Linux and BSD
http://www.remlab.net/miredo/

sixxs1 10 Easy Ministeps to IPv6
http://www.sixxs.net/faq/account/?faq=10steps

sixxs2 IPv4 and IPv6 Distributed Traceroute
http://www.sixxs.net/tools/traceroute/

tbf1 Tunnelbroker.net Forums: Configuring a Tunnel Under Linux
http://www.tunnelbroker.net/forums/index.php?topic=18.0

tdo1 Teredo Overview
https://www.microsoft.com/technet/network/ipv6/teredo.mspx

v4rep IPv4 Address Report (accessed: 2008-06-14)
http://www.potaroo.net/tools/ipv4/index.html

v6ops IPv6 Operators Forum
http://lists.cluenet.de/mailman/listinfo/ipv6-ops