SixXS – Getting IPv6 to the end-user

Jeroen Massar, SixXS
jeroen@unfix.org / jeroen@sixxs.net
SixXS

• Service for providing ISPs with a quick way of enabling their user base with IPv6.
• Tunnel Broker PoPs, provided by various ISPs around Europe (be, ch, ee, fi, de, ie, it, lu, nl, no, pl, pt, si, se, uk), New Zealand and the United States. Thanks to all the ISPs who are providing these PoPs, without them it would not be possible.
• FAQ, Wiki and Forum.
• 15.000++ active users and tunnels.
• 7000++ active subnets (/48’s).
• Only two people running it: Pim & Jeroen
• With the help of all the people at the ISPs who make sure the IPv4+IPv6 routing works to their PoPs, thanks to them for providing this service
Overview:

• Started in 1999 by Pim van Pelt
• 1 PoP: p200, 64mb memory @ AMS-IX
• Static 6bone /127 Tunnels & /60 subnets
• Mostly shell scripts (read: bash)
• MySQL database
• Monitoring using customized fping & rrdtool
• About 200 users
IPng.nl

Problems:

• Scalability problems (management, stats)
• Only static tunnels, thus not useful for dialup/cable
• Too difficult for non-technical users
• Messy code (according to Pim ;)
• Again: too much work to maintain in general
Protocol 41

- Protocol 41 = IPv6
- It specifies how to put an IPv6 packet directly inside an IPv4 packet.
- Protocol 41 is static only.
- Protocol 41 doesn’t cross NATs.
- Protocol 41 is ‘hacker’ according to some:
  - Firewalls see this, user gets an alert of ‘weird’ traffic, user yells abuse.
  - This can happen when a ‘static’ endpoint is not so static, eg cable links, dialup, etc.
RFC3053 – IPv6 Tunnel Broker

Tunnel Broker

config

PoP

IPv6 router

server

IPv4 router

you
SixXS v1

• New idea: a distributed Tunnel Broker
  – 1 IPv6 prefix (eg a 6bone /32)
  – 1 IPv4 address per PoP
  – Allow users to connect to multiple PoPs
  – Do routing
  – We called it: Six Access -> SixXS
    (thanks to XS4all.nl for the ‘XS’ part)
SixXS v1

• Several problems with that though:
  – ISP’s reluctant to carry traffic for others
  – Requires cross-tunnelling between PoPs
  – Different OS’s on different PoPs, synchronisation of statistics and management in general becomes quite hard
  – Introduces new places where things can break
  – More difficult for the user in the end and most likely doesn’t give them any advantage (except when a PoP goes down)

• Additionally time issues caused this idea to be aborted… thus…
SixXS v2

- Pim and Jeroen as core team.
- Second go at it, thus v2.
- Design a TB from the ground up.
- White label, so we could ‘brand’ it.
- Central control site & Multiple PoPs
- Shell & Web interface for the backend
- PHP as the common language
- Credit system to limit abuse
- Launched at AMS-IX Awareness Day (AIAD)
SixXS v2 Toolkit
v2 was still not ‘automatic’ enough:
- No direct updates
- Sometimes synchronizations where missed, manual changes or other routing daemons on the PoP would cause changes which the scripts didn’t handle.
SixXS v3

- Thus v3: C-based daemons on the PoPs
- Kept using our sync/configuration architecture which was designed with such a change in mind
- Act as a routing daemon and “counteract” changes where needed
- TIC support, AICCU to use it
- AYIYA support
- Better logging, notification and tracking
SixXS Tunnel Broker
Heartbeat

- Dynamic/non-24/7 IPv4 endpoints.
- Proto-41 is static. The moment the user unplugs, another user can get that IPv4 address. That user then gets proto-41 packets and the firewall tool beeps with warnings, which sometimes results in abuse reports because we are attacking them.
- Allows one to move around proto-41 tunnels automatically or enable/disable them on the fly.
AYIYA – Anything in Anything

- Proto-41 tunnels can’t cross NATs.
- Proto-41 tunnels are not authenticated.
  (read: one can spoof them easily)
- Heartbeat runs next-to the proto-41 tunnel.
  Heartbeat might work, proto-41 might not.

AYIYA solves these issues by tunneling IPv6 inside IPv4/UDP and signing these packets.
Tunnel Information & Control

- Short: TIC
- Simple SMTP-alike command/answer protocol
- Authentication
- Tunnel Configuration Information
- Tunnel Modification
- Route/Subnet Information
AICCU

“Automatic IPv6 Connectivity Client Utility”

• Proto-41, heartbeat and AYIYA tunnels.

• Windows GUI, Debian Debconf, CLI.

• More-or-less click-and-have-IPv6
Fritz!Box IPv6

  
  “New: FRITZ! IPv6 Lab
  For FRITZ!Box Fon WLAN 7270 or 7570 VDSL
  Features at a glance
  
  ....
  
  IPv6 – SixXS Tunnel Broker support”

Does TIC + heartbeat + subnet delegation
Configureable from the webinterface,
Just enter the user login/password
Abuse handling

• Lots of people seem to just want to abuse IPv6: ability to ‘hide’ on IRC and then annoy people, inviting an DoS which ends up at the PoP.

• We thus require proper details at signup, and verify these. Non-free email addresses solves already a large portion of this as getting a new ISP address is difficult, and very hard when your ISP knows about ones misbehavior.

• Good contacts with various IRC network where the abuse generally comes from
Common Issues

- AICCU takes quite a few setup problems away, but it is far from perfect as it can’t solve broken systems.
- DNS issues (AAAA being dropped)
- Firewalling tools just dropping and thus breaking IPv6.
- Misconfigured Firewalls
- Mis-understanding
Management

```
jeroen@noc-sioss.net - PuTTY

Usage: admin [user|tunnel]route|pop|mail|dnas|review] {help|args}
jeroen@noc-sioss.net - PuTTY
Usage: user {create|update|delete|verify|approve} NICHD
user checkwhois NICHD {report}
user show|view NICHD {full}
user {showlog|viewlog|viewdata}[ip] [[R]NICHD|DATE] [LINKS][all]
user list [(name|email|state|address|country|maintainer|ip) QUERY [exact]]
user {modify|allow|deny|remove|usercredit|allow|deny} NICHD {yes|no}
user setstate NICHD {enabled|disabled|deleted}
user setadmin NICHD {yes|no}
user setcountry NICHD COUNTRYID
user setpass NICHD <password>
user credit NICHD <integer> [reason]
user log NICHD MESSAGE
user popright NICHD POPNAME {A[llow]|D[eny]|R[emove]}
user getemail NICHD
user rename OLDNICHD NEWNICHD
user forum posts NICHD

jeroen@noc-sioss.net - PuTTY
Usage: tunnel request NICHD POPNAME IPv4|IPv6 CITY COUNTRYID
usage: tunnel {delete|show} TUNID
usage: tunnel list [ippop|user|ipv4|ipv6]... [exact]
usage: tunnel move TUNID {NEWIPv4|IPv6|heartbeats|ayya}
usage: tunnel setstate TUNID admin {enabled|disabled|requested}
usage: tunnel setstate TUNID user {enabled|disabled|deleted}
usage: tunnel approve TUNID [POPNAME | POPPREFIXNAME]
usage: tunnel {showlatency} TUNID
usage: tunnel set TUNID {country|city|heartbeat|password|private|mtu} VALUE
usage: tunnel chown TUNID <NICHD>

jeroen@noc-sioss.net - PuTTY
Usage: route request TUNID DESCRIPTION
usage: route list [pop|user|prefix|tunnel] QUERY exact
usage: route setstate ROUTEID {admin|user} {enabled|disabled|deleted|requested}
usage: route settype TUNID {static|bgp}
usage: route approve ROUTEID [POPREFIXNAME] [PREFIX]
usage: route {delegate|undelegate} ROUTEID NAMERVER
```

Jeroen Massar – NANOG 48
Reviewing

As most faulty requests have similar things wrong we have a standard list of rejections, thus don’t be offended when you get rejected, it is not only you…
IPv6Gate

Allows access to any IPv4 website over IPv6 from IPv6-only hosts.
http://www.google.com.sixxs.org
(or get in their approved DNS list ;)

Also allows the reverse: IPv6-only site from IPv4-only host:
GRH – Ghost Route Hunter

- Peers actively with over 150 ISPs around the world.
- A tool for detecting and hunting down Ghost Routes in the IPv6 routing tables and displaying DFP availability.
- Distributed Looking Glass
- Missing Prefixes
- Prefix Comparison
- [http://www.sixxs.net/tools/grh/](http://www.sixxs.net/tools/grh/)
  (See also possible lightning talk about this)
RFC4193 - ULA

IPv6 ULA (Unique Local Address)
RFC4193 Registration

- fd00::/8 ULA Locally Assigned. It is Unique, but maybe not Unique enough as it has a chance that it is not.
- fc00::/8 ULA “Registered” – not specified and thus can’t be used.
- Nearly 200 registrations
- Of course not guaranteed, when people don’t check this list it can’t be.
Future / Wish list

• New signup mechanics
• Multicast
  – Most PoPs already in the SixXS Multicast cloud, but need more testing/experiments
• AYIYA/DNS and AYIYA/HTTP(S)
• New AICCU client
• BGP Support / Multi-PoP Tunnels
• Community Edition
• More and more and more and more PoPs!

http://www.sixxs.net/about/technology/
SixXS for ISPs

First the ISP contacts popmaster@sixxs.net and discuss the possible PoP with us.

The ISP provides:
• A policy describing who can use their PoP
• A (virtual) host
• 1 IPv4 + 2 IPv6 addresses (management.
• Routes IPv6 /40 or more to it
• Installs our root SSH key
• Delegate the reverse for the prefix to us.
• Register the prefix in RIPE
ISP gets

- A PoP for their users (eg their paying customers, or users in country/region X etc) providing all these folks with IPv6
- We handle signups, tunnel and subnet requests. Optionally this can be automated with a backend to the ISP, eg POP/IMAP-based authentication, ACL or other methods.
- Experience in IPv6 in the core
- For free, as SixXS is a project for the community, getting IPv6 to the world is the ultimate goal (along with having to shutdown the IPv6 tunnel broker because everybody has IPv6).
Questions?

Jeroen Massar

JRM1-RIPE

http://unfix.org/~jeroen/  http://www.sixxs.net/

jeroen@unfix.org  jeroen@sixxs.net