NAME

gen6dns — generate IPv6 DNS address and reverse records

SYNOPSYS

gen6dns {**−h**|**−V**}

gen6dns −R [**−b** <*bits*>] *prefix*|*hostname* ...

gen6dns [*options*] [**−f**] [**−s**|**−S**] {**−p** *prefix* |**-6** *ipv4addr* ...} *inputfile* ...

gen6dns [*options*] [**−r**] [**−b** <*bits*>] [**−o** *origin*] {**−p** *prefix* |**-6** *ipv4addr* ...} *inputfile* ...

gen6dns [*options*] **−d** [**−o** *origin*] {**−p** *prefix* ...} {**−P** *prefix* ...} *inputfile* ...

DESCRIPTION

With *gen6dns* it is easy to generate forward and reverse DNS records for IPv6 hosts. If you have a list of hostnames and mac addresses where each host is assigned to a subnet, the command will generate AAAA and PTR records for all the hosts.
It is possible and intended to use several provider prefixes to generate more than one AAAA record per host. Additionally *gen6dns* is able to use scope identifier to support split DNS configurations with different views.

A simple example will be a network with one subnet and a couple of hosts in it:

```
#name   int_id / mac_address   subnet_id

ipad    00:17:53:85:80:3b      :1:  # ipad uses a mac address based id
nas     0018.37ac.7801         :1:  # same for nas, but different syntax
laptop  ::d9b2:56f3:7694:1c5c  :1:  # random static interface id (MS)

rtr-int 00:13:35:a2:91:f3      :1:  # rtr has two interfaces (subnet 1 and 0)
rtr-ext ::1                    :0:  # manually set interface id
```

The interface identifier field can be specified as a MAC address or as an (up to 64 bit long) interface identifier starting with two colons. The subnet id is written as a hex digit string with up to 4 digits (16 Bit) enclosed in colon characters.
Now you can run gen6dns to generate a list of AAAA records with a given provider prefix:

gen6dns -f -S -p 2001:db8:b5b1:ab00::/56 hosts.txt

```
ipad           IN  AAAA   2001:db8:b5b1:ab01:217:53ff:fe85:803b
nas            IN  AAAA   2001:db8:b5b1:ab01:218:37ff:feac:7801
laptop         IN  AAAA   2001:db8:b5b1:ab01:d9b2:56f3:7694:1c5c
rtr-int        IN  AAAA   2001:db8:b5b1:ab01:213:35ff:fea2:91f3
rtr-ext        IN  AAAA   2001:db8:b5b1:ab00::1
```

You can also generate a list of PTR records out of the same input file:

gen6dns -r -o example.com -p 2001:db8:b5b1:ab00::/56 hosts.txt

```
b.3.0.8.5.8.e.f.f.f.3.5.7.1.2.0.1.0    IN  PTR   ipad.example.com.
1.0.8.7.c.a.e.f.f.f.7.3.8.1.2.0.1.0    IN  PTR   nas.example.com.
c.5.c.1.4.9.6.7.3.f.6.5.2.b.9.d.1.0    IN  PTR   laptop.example.com.
3.f.1.9.2.a.e.f.f.f.5.3.3.1.2.0.1.0    IN  PTR   rtr-int.example.com.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0    IN  PTR   rtr-ext.example.com.
```

GENERAL OPTIONS

−V, **−−version**
Output version information and exit

−h, **−−help**
Print a short description of the command options.

−C *char***, −−comment=***char*
Use *char* as comment character instead of a '#' which is the default.

−D *char***, −−delim=***char*
Add *char* to the list of delimiter chars. The default delimiter list is space and tab.

−v, **−−verbose**
Print some debug messages on stderr. Use twice to get the output of a DEBUG directive in file parsing.

−w, **−−write**
Force to write output into files instead of printing it on stdout. If this flag is given, forward records will be written to files named `g6d.`*domain* or `g6d_`*view*`.`*domain*.
Reverse records will be written to files named `g6r.`*revbit* or `g6r_`*view*`.`*revbit*.

Depending on the prefix size and the given bitmask (option **−b)** reverse zones are split of at that boundary which likly results in a lot of generated files. For example, if you have 1000 subnets and set **-b 64** then in theory 1000 reverse files will be generated. Actually the maximum number of files is limited by a compile time constant which is by default set to 128. If you really want to get 1000 files like in the example above, you have to change MAXFILEP and re-compile gen6dns.

−a, **−−append**
Like **−w** but does not overwrite the output files. Instead all output will appended to the existing file.

−t *ttlspec***, −−ttl=***ttlspec*
With this option, all records will get a TTL time in seconds added. The *ttlspec* is an integer number, optionally followed by an unit given as a single character, where **s** are seconds, **m** denotes minutes, **h** hours, **d** days and **w** weeks.

MODE SPECIFIC OPTIONS

Some of the options are only available or useful in case of generating forward DNS records, while others are only useful for reverse record generation and some are used for dynamic dns updates.

−R, **−−revzone**
With this option the ip6.arpa reverse zone for the literal prefix given as argument is printed on stdout. If the argument is a hostname, a dns lookup for an AAAA record is made and a PTR record with the ipv6 address as label is printed on stdout. The option *-b* can be used to change the label to be relative to the size of the reverse zone, which is usually on subnet (64 bit) or on prefix (/60, /56 or /48) boundary.

```
$ gen6dns -R 2001:db8:abc::/48
c.b.a.0.8.b.d.0.1.0.0.2.ip6.arpa.

$ gen6dns -R -b 64 horst.example.net
8.1.0.0.c.3.c.8.2.a.f.e.f.c.e.3  IN  PTR   horst.example.net.  \
                              ; 2001:db8:abc:0:5e26:aff:fe73:cc44
```

−f, **−−forward**
Generate forward (AAAA) records only. If neither **−f** nor **−r** is given, than both type of DNS records are generated.

−s, **−−squeeze**
: Print IPv6 addresses in a compact form where leading zeros within a block of hex digits is suppressed. This option is only useful in forward mode.

−S, **−−full-squeeze**
: Print IPv6 addresses in a more compact form where not only leading zeros within a block of hex digits are suppressed, but also a continuous list of zero blocks are written as two colons (e.g. 2001:db8:1::15). This option is only useful in forward mode.

−r, **−−reverse**
: Generate reverse (PTR) records only. If neither **−f** nor **−r** is given, than both type of DNS records are generated.

−d, **−−ddns**
: With this option dynamic update add/delete commands will be written on stdout. This option works only in forward mode so this is turned on automatically. For all prefixes specified by a -p (--add-prefix) option an update add line is generated, while for all -P (--del-prefix) options an update delete line is printed of stdout. The output is suitable to pipe it to nsupdate(1).
Of course, the forward zone must be a dynamic one.

−l *name***, −−lookup=***name*
: Print out update add/del lines to stdout only for hosts matching <name>. This is useful if a new host has to be added to a dynamic zone (use -p option), or to delete a host from the zone (use option -P in this case). This option can be used in dynamic mode only.

−b *mask***, −−bits=***mask*
: Split reverse zones on *mask* boundary. Be aware that this could result in a lot of files created (see Option −w). This option is only useful in reverse mode.

−o *domain***, −−origin=***domain*
: Define the domain added to the rdata of a PTR record (default is example.net.). This option is only useful in reverse mode.

−p *prefix***, −−add-prefix=***prefix*
: This option adds an IPv6 prefix to the list of prefixes used to generate AAAA records and can be given more than once.
If a scope is defined in the config file, the given prefix will matched against the scope list and records will be generated only for those hosts matching the scope list.

While this option is only necessary in forward mode, it helps also in reverse mode to find the zone cut for reverse zone file generation.

−P *prefix***, −−del-prefix=***prefix*
: This option deletes an IPv6 prefix from a dynamic zone, so it is only useful in combination with option -d (--ddns).

−6 *ipv4-addr***, −−6to4=***ipv4-addr*
: This works like option **−p** but the IPv6 prefix used is build out of the 6to4 prefix with the given IPv4 address added (e.g. -6 1.2.3.4 will be the same as -p 2002:0102:0304::/48). Be aware that RFC7526 deprecates the use of anycast prefix for 6to4 relay routers which means that 6to4 is no longer useful as a general transition mechanism.

SAMPLE USAGE

Take a look at the example directory of the source tree for examples to use **gen6dns** in different secenarios (residential user, SOHO network).

FILES

Subnet definition file

A file with a %%SUBNET SECTION to map subnet names to subnet identifier.

The subnet id is a 16 bit hex value enclosed in colons, representing the bits 48 to 64 of an IPv6 address. Depending on the prefix size not all of the subnet id bits will be used to create the subnet prefix.

In the following example, if the IPv6 prefix on the command line will be just a /56 (2001:db8:1:ef00::/56), only the righmost 8 bits of the subnet id will be used. So "subnet1" will result in 2001:db8:1:efb1::/64 and "subnet2" in 2001:db8:1:efb2::/64.

```
%%SUBNET SECTION
#name subnetid

subnet1     :0ab1:
subnet2     :0ab2:
subnet3     :0001:
```

scope definition file

A file with a %%SCOPE SECTION to map IPv6 prefixes to scope parameter. It's primary use is to put ULA names into a different view than into the one used for global IPv6 addresses.
The name of the scope is referenced in a host definition. The 'view' string is used to put the resource records in separate files suitable named (see option **−w**). The string "*" or "none" is used to not specify a dedicated view (default).
The domain is used for the generation of reverse PTR records instead of the domain given as option **−o**. The matching prefix entry is to distinguish between one or the other scope.

```
#
#     A file for scope definitions (domain names, views, and matching prefix)
#

%%SCOPE SECTION
#name       view  domain                  matchprefix

prov1       *     example.de.       2003::/19   # telekom
prov2       *     example.de.       2a00::/22   # vodafone (arcor)

glob        none  example.de.       2000::/3    # match on any public prefix
ula         intern      mgmt.example.de.  fd00::/8
```

host definition file

A file with an optional %%HOST SECTION to map host (interface) names to interface identifiers and subnet id's.

```
#
#     A file with host names, IID (mac address or interface identifier)
#     and related subnet id
#

%% HOST SECTION
#name       int_id / mac_address    subnet_id   scope

horst       00:17:53:85:80:3b clt         [prov1, prov2]
ns1         ::53              srv         [prov1]
ns2         ::d9b2:56f3:7694:1c5c   srv         [prov2]

hugo <24h>  00:13:35:a2:91:f4 :1:         [prov1, prov2, mgmt]
ˆ           IN  A       1.2.3.4
ˆ           IN  SSHFP   1 1   12345637339379337730723702742724 27e8=
ˆ           IN  SSHFP   1 2   533a56373393793377307237027427242 7e8=

gustav.test 0013.35a2.91f5          :1:         [mgmt]

rtr1        ::100b:0:0:1            lo          [mgmt]
```

The name of the host is the first text in a line. If it is a multi homed host than it is the name of one of the interfaces of this host. The name can be a subdomain (e.g. host.ext) and optionally followed by a TTL specification enclosed in angle brackets.
If the name is a caret, than the line is a continuation of the previous host entry, and the line itself will be copied to the forward file unmodified.

The next field is the interface identifier specified as the rightmost 64 bit of the IPv6 address, or as a 48 bit ethernet mac address. In the last case the usual way to write down a mac address is supported (colon or dashed seperated six bytes or dot seperated two byte hex values). In the former case the string must start with two colons and must be specified as a 64 bit IPv6 address which could abbreviated in the usual way.

The next field is the subnet ID written as subnet id name or as the subnet id number (enclosed in colons).

The last field is the list of scope for this host entry. The list is enclosed in square brackets with each entry delimited by comma.

BUGS

Some of the options are only meaningful in certain gen6dns modes.

A subnet specific scope would be useful too.

There are some debug commands implemented in the file parser which helps in debugging the internal data structures.

```
%%DEBUG print-scopelist
%%DEBUG print-subnetlist
%%DEBUG print-prefixlist
```

These commands should not be used during normal operation.

The caret syntax of an host entry is not supported in dynamic update mode. However, this is an ugly hack anyway and will not supported in upcoming versions of gen6dns.

AUTHORS

Written by Holger Zuleger

COPYRIGHT

SEE ALSO

named(8), rndc(8), soaserial(1), dig(1)