NAME

dnsdbq — DNSDB query tool

SYNOPSIS

dnsdbq [**-cdfgGhIjmqSsUv468**][**-A** *timestamp*][**-B** *timestamp*][**-b** *bailiwick*]
[**-i** *ip*][**-J** *input_file*][**-k** *sort_keys*][**-L** *output_limit*]
[**-l** *query_limit*][**-M** *max_count*][**-N** *raw_name*][**-n** *name*][**-O** *offset*]
[**-p** *output_type*][**-R** *raw_rrset*][**-r** *rrset*][**-t** *rrtype*]
[**-u** *server_sys*][**-V** *verb*]

DESCRIPTION

dnsdbq constructs and issues queries to Passive DNS systems which return data in the IETF Passive DNS Common Output Format. Farsight Security's DNSDB is one such system. **dnsdbq** displays responses in various formats. It is commonly used as a production command line interface to such systems.

Its default query type is a "lookup" query. As an option, it can issue a "summarize" query type.

Farsight Security's DNSDB system implements both APIv1 and APIv2 interfaces. APIv1 is accessed by specifying system "dnsdb." APIv2 is accessed by specifying system "dnsdb2".

You'll need to get an API key from Farsight to use **dnsdbq** with DNSDB.

Farsight's passive DNS infrastructure performs a complex process of "bailiwick reconstruction" where an RRset's position within the DNS hierarchy is approximated. This serves two purposes:

1. Provide context of the location of a given DNS record within the DNS hierarchy
2. Prevent "untrustworthy" records that are a result of intentional or unintentional cache poisoning attempts from being replicated by downstream consumers.

For example, given the fully qualified domain name **www.dachshund.example.com**, valid bailiwicks would be **dachshund.example.com**, **example.com**, or **com**.

OPTIONS

-A *timestamp*
Specify a backward time fence. Only results seen by the passive DNS on or after this time will be selected. See also **-c**. See the TIMESTAMP FORMATS section for more information about this.

-B *timestamp*
Specify a forward time fence. Only results seen by the passive DNS sensor network on or before this time will be selected. See also **-c**. See the TIMESTAMP FORMATS section for more information about this.

-b *bailiwick*
specify bailiwick (only valid with **-r** queries).

-c by default, **-A** and **-B** (separately or together) will select partial overlaps of database tuples and time search criteria. To match only complete overlaps, add the -c ("completeness") command line option (this is also known as "strict" mode).

-d enable debug mode. Repeat for more debug output.

-f specify batch lookup mode allowing one or more queries to be performed. Queries will be read from standard input and are expected to be in one of the following formats:

- RRset (raw) query: **rrset/name/NAME[/RRTYPE[/BAILIWICK]]**
- RRset (raw) query: **rrset/raw/HEX[/RRTYPE[/BAILIWICK]]**
- Rdata (name) query: **rdata/name/NAME[/RRTYPE]**
- Rdata (IP address) query: **rdata/ip/ADDR[,PFXLEN]**

- Rdata (raw) query: **rdata/raw/HEX[/RRTYPE]**
- Change query options: **$OPTIONS {options}**

Where **options** :==

```
[-A timestamp] [-B timestamp] [-c] [-g] [-G]
[-l query_limit] [-L output_limit] [-O offset]
```

$OPTIONS alone on a line allows command line options to be changed mid-batch. If no options are given, the query parameters will be reset to those given on the command line, if any, or else to defaults.

A line starting with a # will be ignored as a comment.

Any internal slash (/) or comma (,) characters within the search names of a batch entry must be URL-encoded (for example, %2F or %2C).

For raw queries, the HEX value is an even number of hexadecimal digits specifying a raw octet string. The "raw" wire-format encodings are standardized. The embedding of these in dnstable is documented in the `dnstable-encoding`(5) manual page.

In batch lookup mode, each answer will be followed by a -- marker, so that programmatic users will know when it is safe to send the next lookup, or if lookups are pipelined, to know when one answer has ended and another begun. This option cannot be mixed with **-n**, **-r**, **-R**, or **-i**. See the EXAMPLES section for more information on how to use **-f**.

If two **-f** options are given, then each answer will also be preceded by a ++ marker giving the query string (as read from the batch input) in order to identify each answer when a very large batch input is given, and the -- marker will include an error/noerror indicator and a short message describing the outcome. With two **-f** options and also **-m**, answers can appear in a different order than the batched questions.

The ++ and -- markers are not valid JSON, CSV, or DNS (text) format, so caution is required. (See **-m** option below.)

-g return graveled results. Default is to return aggregated results (rocks, vs. gravel). Gravel is a feature for providing Volume Across Time.

-G undo the effect of **-g**, this returning rocks rather than gravel. (Used in $OPTIONS in batch files.)

-h emit usage and quit.

-I request information from the API server concerning the API key itself, which may include rate limit, query quota, query allowance, or privilege levels; the output format and content is dependent on the server_sys argument (see **-u**) and upon the **-p** argument. **-I -p json** prints the raw info. **-I -p text** prints the information in a more understandable textual form, including converting any epoch integer times into UTC formatted times.

-i *ip*
specify rdata ip ("right-hand side") query. The value is one of an IPv4 address, an IPv6 address, an IPv4 network with prefix length, an IPv4 address range, or an IPv6 network with prefix length. If a network lookup is being performed, the delimiter between network address and prefix length is a single comma (",") character rather than the usual slash ("/") character to avoid clashing with the HTTP URI path name separator. See EXAMPLES section for more information about separator substitution rules.

-J *input_file*
opens input_file and reads newline-separated JSON objects therefrom, in preference to -f (batch mode) or query mode. This can be used to reprocess the output from a prior invocation which used **-j** (-p json). Sorting, limits, and time fences will work. Specification of a domain name, RRtype, Rdata, or offset is not supported at this time. If input_file is "-" then standard input (stdin) will be read.

-j specify newline delimited json output mode.

-k *sort_keys*
when sorting with -s or -S, selects one or more comma separated sort keys, among "first", "last", "duration", "count", "name", and/or "data". The default order is be "first,last,duration,count,name,data" (if sorting is requested.) Names are sorted right to left (by TLD then 2LD etc). Data is sorted either by name if present, or else by numeric value (e.g., for A and AAAA RRsets.) Several **-k** options can be given after different **-s** and **-S** options, to sort in ascending order for some keys, descending for others.

-l *query_limit*
query for that limit's number of responses. If specified as 0 then the DNSDB API server will return the maximum limit of results allowed. If **-l**, is not specified, then the query will not specify a limit, and the DNSDB API server may use its default limit.

-L *output_limit*
clamps the number of objects per response (under **-[R|r|N|n|i|f]**) or for all responses (under **-[fm|ff|ffm]**) output to **output_limit**. If unset, and if batch and merge modes have not been selected with the **-f** and **-m** options, then the **-L** output limit defaults to the **-l** limit's value. Otherwise the default is no output limit.

-M *max_count*
for the summarize verb, stops summarizing when the count reaches that max_count, which must be a positive integer. The resulting total count may exceed max_count as it will include the entire count from the last rrset examined. The default is to not constrain the maximum count. The number of rrsets summarized is also limited by the query_limit.

-m used only with **-f**, this causes multiple (up to ten) API queries to execute in parallel. In this mode there will be no "--" marker, and the combined output of all queries is what will be subject to sorting, if any. If two **-f** flags are specified with **-m**, the output will not be merged, can appear in any order, will be sorted separately for each response, and will have normal '--' / '++' markers. (See **-f** option above.)

-N *HEX*
specify raw **rdata** data ("right-hand side") query. HEX is as described above.

-n *name*
specify **rdata** name ("right-hand side") query. The value is a DNS domain name in presentation format, or a left-hand (".example.com") or right-hand ("www.example.") wildcard domain name. Note that left-hand wildcard queries are somewhat more expensive than right-hand wildcard queries.

-O *offset*
to offset by #offset the results returned by the query. This gives you incremental results transfers. Cannot be negative. The default is 0.

-p *output_type*
select output type. Specify:

text for presentation output meant to be human-readable. This is the default. **dns** is a synonym, for compatibility with older programmatic callers.

json for newline delimited JSON output.

csv for comma separated value output. This format is information losing, since it cannot express multiple resource records that are in a single RRset. Instead, each resource record is expressed in a separate line of output.

See the **DNSDB_TIME_FORMAT** environment variable below for controlling how human readable timestamps are formatted.

-q makes the program reticent about warnings.

-R *HEX*
specify raw **rrset** owner data ("left-hand side") query. HEX is as described above.

-r *rrset*
specify rrset ("left-hand side") name query.

-s sort output in ascending key order. Limits (if any) specified by **-l** and **-L** will be applied before and after sorting, respectively. In batch mode, the **-f**, **-ff**, and **-ffm** option sets will cause each batch entry's result to be sorted independently, whereas with **-fm**, all outputs will be combined before sorting. This means with **-fm** there will be no output until after the last batch entry has been processed, due to store and forward by the sort process.

-S sort output in descending key order. See discussion for **-s** above.

-t *rrtype*
specify the resource record type desired. Default is ANY. If present, this option should precede any **-R**, **-r**, **-N**, or **-n** options. This option is not allowed if the **-i** option is present. Valid values include those defined in DNS RFCs, including ANY. A special-case supported in DNSDB is ANY-DNSSEC, which matches on DS, RRSIG, NSEC, DNSKEY, NSEC3, NSEC3PARAM, and DLV resource record types.

-u *server_sys*
specifies the syntax of the RESTful URL, default is "dnsdb".

-V *verb*
The verb to perform, i.e. the type of query, either "lookup" or "summarize". The default is the "lookup" verb. As an option, you can specify the "summarize" verb, which gives you an estimate of result size. At-a-glance, it provides information on when a given domain name, IP address or other DNS asset was first-seen and last-seen by the global sensor network, as well as the total observation count.

-U turns off TLS certificate verification (unsafe).

-v report the version of dnsdbq and exit.

-4 use to force connecting to the DNSDB server via IPv4.

-6 use to force connecting to the DNSDB server via IPv6.

-8 Normally dnsdbq requires that **-n** or **-r** arguments are 7-bit ASCII clean. Non-ASCII values should be queried using PUNYCODE IDN encoding. This **-8** option allows using arbitrary 8 bit values.

TIMESTAMP FORMATS

Timestamps may be one of following forms.

- positive unsigned integer : in Unix epoch format.
- negative unsigned integer : negative offset in seconds from now.
- YYYY-MM-DD [HH:MM:SS] : in absolute form, in UTC time, as DNSDB does its fencing using UTC time.
- %uw%ud%uh%um%us : the relative form with explicit labels (w=weeks, d=days, h=hours, m=minutes, s=seconds). Calculates offset from UTC time, as DNSDB does its fencing using UTC time.

When using batch mode with the second or forth cases, using relative times to now, the value for "now" is set when dnsdbq starts.

A few examples of how to use timefencing options.

```
# only responses after Aug 22, 2015 (midnight)
$ dnsdbq ... -A 2015-08-22
# only responses before Jan 22, 2013 (midnight)
$ dnsdbq ... -B 2013-01-22
# only responses from 2015 (midnight to midnight)
$ dnsdbq ... -B 2016-01-01 -A 2015-01-01
# only responses after 2015-08-22 14:36:10
$ dnsdbq ... -A "2015-08-22 14:36:10"
# only responses from the last 60 minutes
$ dnsdbq ... -A "-3600"
# only responses after "just now"
$ dnsdbq -f ... -A "-3600"
# batch mode with only responses after "just now", even if feeding inputs
to dnsdbq in batch mode takes hours.
$ date +%s
1485284066
$ dnsdbq ... -A 1485284066
```

EXAMPLES

A few examples of how to specify IP address information.

```
# specify a single IPv4 address
$ dnsdbq ... -i 128.223.32.35
# specify an IPv4 CIDR
$ dnsdbq ... -i 128.223.32.0/24
# specify a range of IPv4 addresses
$ dnsdbq ... -i 128.223.32.0-128.223.32.32
```

Perform an rrset query for a single A record for **farsightsecurity.com**. The output is serialized as JSON and is piped to the **jq** program (a command-line JSON processor) for pretty printing.

```
$ dnsdbq -r farsightsecurity.com/A -l 1 -j | jq .
{
  "count": 6350,
  "time_first": 1380123423,
  "time_last": 1427869045,
  "rrname": "farsightsecurity.com.",
  "rrtype": "A",
  "bailiwick": "farsightsecurity.com.",
  "rdata": [
    "66.160.140.81"
  ]
}
```

Perform a batched operation for a several different **rrset** and **rdata** queries. Output is again serialized as JSON and redirected to a file.

```
$ cat batch.txt
rrset/name/wikipedia.org
rrset/name/dmoz.org
```

```
rrset/raw/0366736902696f00/A
rdata/name/pbs.org
rdata/name/opb.org
rdata/ip/198.35.26.96
rdata/ip/23.21.237.0,24
rdata/raw/0b763d73706631202d616c6c
$ dnsdbq -j -f < batch.txt > batch-output.json
$ head -1 batch-output.json | jq .
{
  "count": 2411,
  "zone_time_first": 1275401003,
  "zone_time_last": 1484841664,
  "rrname": "wikipedia.org.",
  "rrtype": "NS",
  "bailiwick": "org.",
  "rdata": [
    "ns0.wikimedia.org.",
    "ns1.wikimedia.org.",
    "ns2.wikimedia.org."
  ]
}
```

FILES

~/.isc-dnsdb-query.conf, **~/.dnsdb-query.conf**, **/etc/isc-dnsdb-query.conf**, or **/etc/dnsdb-query.conf**: configuration file which should contain the user's apikey and server URL.

`APIKEY` contains the user's apikey (no default).

`DNSDB_SERVER` contains the URL of the DNSDB API server (default is https://api.dnsdb.info), and optionally the URI prefix for the database (default is "/lookup").

`CIRCL_AUTH`, `CIRCL_SERVER` enable access to a passive DNS system compatible with the CIRCL.LU system.

`DNSDBQ_SYSTEM` contains the default value for the *u* option described above. Can be "dnsdb", "dnsdb2", or "circl". If unset, **dnsdbq** will probe for any configured system.

ENVIRONMENT

The following environment variables affect the execution of **dnsdbq**:

`DNSDB_API_KEY`, `APIKEY` contains the user's apikey. If DNSDB_API_KEY is not present, then APIKEY will be used. If neither variable is present, the configuration file is consulted.

`DNSDB_SERVER` contains the URL of the DNSDB API server, and optionally a URI prefix to be used (default is "/lookup"). If not set, the configuration file is consulted.

`DNSDBQ_TIME_FORMAT` controls how human readable date times are displayed. If "iso" then ISO8601 (RFC3339) format is used, for example; "2018-09-06T22:48:00Z". If "csv" then an Excel CSV compatible format is used; for example, "2018-09-06 22:48:00".

EXIT STATUS

Success (exit status zero) occurs if a connection could be established to the back end database server, even if no records matched the search criteria. Failure (exit status nonzero) occurs if no connection could be established, perhaps due to a network or service failure, or a configuration error such as specifying the wrong server hostname.

SEE ALSO

`dig`(1), `jq`(1), `libcurl`(3), `dnstable-encoding`(5)