Frequently Asked Questions

DNSDB FAQ

Version 24.5

DomainTools

In this Document

S. I) General questions

Q. I-1) What is passive DNS?

Passive DNS uses observed cache miss traffic collected from above recursive resolvers to build a database detailing relationships between domain names, IP addresses, and nameservers. That historical database can then be queried to get a report of:

- Domains that have been seen associated with a particular IP or IP range
- IPs that have been seen associated with a particular domain name
- Domain names that are known to be using a particular authoritative nameserver, etc.
- The date and time range associated with associations and changes

One example of the value of using passive DNS can be seen when you compare what passive DNS finds vs what you may get when you just request a PTR record for that same IP address. For instance, let's request the PTR record for 128.223.142.89:

```
$ dig -x 128.223.142.89 +short
www.uoregon.edu.
```

This PTR makes it appear as if 128.223.142.89 is home to www.uoregon.edu, and at one time perhaps it was. However, at the time this example was prepared, www.uoregon.edu was actually at:

```
$ dig www.uoregon.edu +short
drupal-cluster5.uoregon.edu.
128.223.142.244
```

If we check dnsdb and ask to just see passive DNS records for 128.223.142.89 from the last 6 months (24 weeks), we can see more current results that look like:

```
$ dnsdb_query.py -i 128.223.142.89 --after=24w | sort
cfc.uoregon.edu. IN A 128.223.142.89
culjp.org. IN A 128.223.142.89
oregon-ix.com. IN A 128.223.142.89
oregon-ix.net. IN A 128.223.142.89
oregonix.net. IN A 128.223.142.89
oregonix.org. IN A 128.223.142.89
virt-www.uoregon.edu. IN A 128.223.142.89
www.culjp.org. IN A 128.223.142.89
```

The equivalent command using the dnsdbq CLI command is:

```
dnsdbq -i 128.223.142.89 -A 24w | fgrep -v ';' | grep . | sort
```

Q. I-2) How does passive DNS data differ from WHOIS data?

WHOIS is an online distributed database that documents control over particular Internet resources such as domain names, blocks of IP addresses, and autonomous system numbers (ASNs).

WHOIS normally contains manually-maintained point of contact information, as well as information about the dates when resources were received or modified, plus additional details associated with resources (these details may vary depending on the type of resource or the specific WHOIS operator).

Passive DNS is a database that contains automatically collected information gleaned from DNS queries and responses, and consists of observed and imputed relationships between domain names, IP addresses, and nameservers.

Passive DNS also captures other types of data delivered via DNS, such as DKIM/DMARC records, SPF records, etc.

Q. I-3) How much data is in the DNSDB?

The DNSDB database currently has over 100 billion unique DNS records. We currently see over 200,000 new raw observations/second totaling over 2TB of DNS data collected daily.

Q. I-4) How far back does DNSDB data go? When did you begin collecting and saving data for DNSDB?

While DNSDB's data collection began in 2007, various improvements were made over time. The currently utilized NMSG-based passive DNS architecture was put into production on June 24, 2010, and that is the earliest date you will see for passive DNS data. For example:

```
$ dnsdb_query.py -r www.google.com/cname -s time_first
;;  bailiwick: google.com.
;;      count: 83,954,084
;; first seen: 2010-06-24 04:22:00 -0000
;;  last seen: 2012-09-06 10:49:14 -0000
www.google.com. IN CNAME www.l.google.com.
```

```
[etc]
```

The roughly equivalent command using the dnsdbq CLI command is:

```
$ dnsdbq -r www.google.com/cname -s
```

Some data obtained from ICANN Zone File Access (ZFA) programs may go back slightly further. For example:

```
$ dnsdb_query.py -r google.com/NS/com
;;  bailiwick: com.
;;      count: 2,157
;; first seen in zone file: 2010-04-24 16:12:21 -0000
;;  last seen in zone file: 2016-03-30 16:14:20 -0000
google.com. IN NS ns1.google.com.
google.com. IN NS ns2.google.com.
google.com. IN NS ns3.google.com.
google.com. IN NS ns4.google.com.
```

The equivalent command using the dnsdbq CLI command is:

```
dnsdbq -r google.com/NS/com
```

Q. I-5) I understand that DomainTools collects passive DNS from sensors located all around the Internet. Who specifically contributes data to DomainTools? Do you have any sensors in country X or country Y?

DomainTools has more than 400 sensors deployed Internet-wide, but we do not disclose either the identities of our sensor operators or their locations.

Q. I-6) I'm not seeing some domains that I think I should be seeing in DNSDB. Are you filtering anything out?

Yes, some content is intentionally filtered from DNSDB for operational reasons.

As of July, 2022 DNSDB was changed to reduce the amount of junk wildcard domains in its database. We are gradually rolling out a change to replace multiple wildcarded DNS rrnames with a single rrname that starts with a _WILDCARD_. label. No other rrname labels contain

uppercase letters, so records with this (all upper case) _WILDCARD_. were never in DNSDB before. Note that there are existing, real, domain names that contain a _wildcard_. label (all lower case).

While we do not disclose proprietary details of what we filter, broad categories of content filtered from DNSDB include (but aren't limited to):

- Randomized subdomain attack (garbage) DNS traffic
- Inverse address records for dynamic domain names that match dynamic domain name patterns
- DNS block list query and response traffic
- Busy sites with wildcarded CNAMES (such as blogspot.com)

Q. I-7) I understand that DNSDB adds data to the database as it's seen. What about domains that get created, but never get used? Do you somehow include them, too?

Yes, DomainTools has visibility into domains that have been created but which are not yet used from its participation in the ICANN Zone File Access (ZFA) data sharing program. While we routinely add all domains learned via that ZFA process, the vast majority of DNSDB's data comes from passively observed DNS traffic.

Q. I-8) Can I use DNSDB as a basis for making quantitative estimating about a domain's "popularity" or "importance?"

Because DomainTools collects above the recursive resolver, DomainTools only sees cache miss traffic. The volume of cache miss traffic is largely based on a domain's popularity. Thus, you can at least get a rough sense of a domain's relative popularity, e.g.:

```
$ dnsdb_query.py -r www.google.com/a | grep count | awk '{print $3}' |
\
    sed 's/,//g' | paste -sd+ - | bc
1964126517
```

Vs.

```
$ dnsdb_query.py -r 7o8xg9qm0c.com/a | grep count | awk '{print $3}' |
\
    sed 's/,//g' | paste -sd+ - | bc
2830
```

Obviously www.google.com has been seen far more often than the other relatively-obscure/seemingly-randomly-named domain however an analyst should avoid making hard quantified comparisons (avoid: “domain X is n times more popular than domain Y”)

Q. I-9) Can DNSDB help me ascertain information relevant to DKIM/DMARC and/or SPF?

Yes. DKIM/DMARC and SPF both use the DNS to store the records they need. For example:

```
$ dnsdb_query.py -r \*._domainkey.dmarc.org
;;  bailiwick: dmarc.org.
;;      count: 974
;; first seen: 2012-02-02 07:30:08 -0000
;;  last seen: 2018-10-06 15:19:36 -0000
clochette._domainkey.dmarc.org. IN TXT "v=DKIM1; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCvWWQyy4vbyeNt8YN0KEHfPb5j/
BZHcOD7xu8rPbUoMFD6tskk9kpJOBOlWvei3hx6HWAqa7Q8EzlQc0ijqsRxSgMhvFnYUAK
M2yewGF6+
QVsCPrLal0XvqOF+uAtScBj0BRYvTI9algsH+1DK8VzZ/bvOdoCM3rj8DJ/D8D3ugQIDAQ
AB"
$ dnsdb_query.py -r _dmarc.dmarc.org
;;  bailiwick: dmarc.org.
;;      count: 181
;; first seen: 2012-03-17 19:02:34 -0000
;;  last seen: 2018-09-30 08:04:47 -0000
_dmarc.dmarc.org. IN TXT "v=DMARC1; p=none; pct=100;
rua=mailto:reports@dmarc.org;
    ruf=mailto:reports@dmarc.org"
$ dnsdb_query.py -r dmarc.org/txt | grep spf | sort -u
dmarc.org. IN TXT "v=spf1 a mx -all"
```

Q. I-10) Does DNSDB include AAAA (IPv6) records? DNSSEC records? All IETF-defined record-types?

Unlike some other passive DNS services that may offer only A or AAAA records, DNSDB includes all IETF-defined records types, including all IPv6- and all DNSSEC-related record types.

For example, you can look for Google's AAAA (IPv6) records by specifying:

```
$ dnsdb_query.py -r www.google.com/aaaa
```

You can also easily find reverse IPv6 records in the ip6.arpa domain with PTR records in DNSDB.

When it comes to DNSSEC, you can see the DS records for a sample domain such as internet2.edu by querying:

```
$ dnsdb_query.py -r internet2.edu/ds
```

You can also see DNSKEY records, RRSIG records, and NSEC records in the data.

S. II) Pricing-Related Questions

Q. II-1) How much does DNSDB cost for a "typical" user?

To request a demonstration of DNSDB or to inquire about a trial API key please contact the DomainTools sales team. Some general pricing principles for your background:

- There are three types of quotas: time-based, block-based, and unlimited.
    - Time-based quotas are usually applied on a daily basis and resets daily at 00:00 (midnight) in the UTC time zone. Time-based quotas can also be applied for arbitrary time-quanta, but this is unusual. These quotas are tiered/priced according to maximum daily usage with tiers starting at 1,000 queries/day.
    - Unlimited quotas do not limit the number of queries/day.
- Discounts are available for multi-year prepaid purchases.
- DNSDB Export is also available for "on-premises" deployment for use in offline environments, latency sensitive environments, or circumstances where special access to the DNSDB data is required.
- Because DNSDB has security-sensitive information, all customers must be pre-approved for access. DomainTools reserves the right to decline any potential customer at its sole discretion.

Q. II-2) You mentioned that for time-based quotas, users contract for a maximum number of DNSDB queries per day. Do unused DNSDB queries "carry forward" or "roll over" for use in subsequent days?

No. The query volume tier you purchase is essentially a "reservation of capacity" on our infrastructure.

Q. II-3) For block-based quotas, what happens if there are still unused queries left when the quota expires?

Unless renewed prior to expiration, when a block-based quota expires, unused queries are lost. If the Block Quota Subscription is renewed PRIOR to expiration, unused queries will be added to the new subscription quota.

Q. II-4) If I purchase an "unlimited" license for DNSDB, does that mean I can open hundreds of parallel (concurrent) query streams?

No, all customers are limited to a maximum of 10 parallel query streams at one time.

If more parallel (concurrent) query streams are required for your use case, let's discuss DNSDB Export as an option.

Q. II-5) I run busy recursive resolvers. If I contribute data from them, can I get a discount from DomainTools?

Yes. Discount levels are based on the value of the contributions, measured by volume and uniqueness of the data shared. In a few cases, partners who have shared substantial volumes of unique data (such as large ISPs) have been eligible for 100% discounts.

Q. II-6) I'd like to contribute data. How do I install and run a sensor?

If you would like to contribute data, please contact us and let us know. The entire DomainTools community would be delighted.

DomainTools Passive DNS collects DNS response data received by caching, recursive DNS servers distributed around the Internet. This data is aggregated and made available via the DomainTools Security Information Exchange platform where it is imported in an anonymized form into DomainTools DNSDB. Operating a DomainTools Passive DNS sensor improves the quality of data available from DomainTools DNSDB and aids anti-abuse research.

The passive DNS sensor only collects the DNS data received by a caching server as the result of recursion. The queries sent by individual clients are never logged. The sensor also offers the ability to zero out the IP address of the resolver.

Q. II-7) I run a farm of busy authoritative nameservers. Can I contribute data to DNSDB and get a discount?

We do participate in ICANN's Zone File Access (ZFA) program, but don't currently collect data from authoritative nameserver operators. If you operate authoritative nameservers and would like to discuss data sharing opportunities, please get in touch with DomainTools.

Q. II-8) I'm a university faculty member or graduate student. Do you have discounted or free access to DNSDB for academic researchers?

DomainTools enthusiastically supports academic research, and is happy to consider requests for discounted or free access to DNSDB. Please contact us with your request and we will evaluate it to see what's possible.

Q. II-9) I fight cyber crime on a volunteer basis (no pay or other compensation, just working for the good of the Internet). Can I obtain discounted or free access to DNSDB?

DomainTools is pleased to support bona-fide "do-gooders" working to better the Internet by offering deeply discounted or free access to DNSDB. Please contact us with your request and we will evaluate it to see what's possible.

S. III) Privacy-Related Questions

Q. III-1) Where are DomainTools' facilities physically located? What about DomainTools sensor operators and customers?

DomainTools is headquartered in San Mateo, California, USA. Our data centers are located in the states of Virginia and California of the USA. We have customers and sensor operators distributed internationally, in addition to numerous domestic customers and sensor operators.

Q. III-2) We'd like to contribute passive DNS data, but need to understand how you protect end user privacy and sensor operator identities. What steps do you take?

Because DomainTools collects cache miss traffic from above large recursive resolvers, query traffic appears to come from the recursive resolvers themselves rather than any individual user. This architecture provides substantial privacy protection for end users at sites that contribute data to DNSDB.

As a security policy matter, DomainTools does not disclose the identity of DomainTools' sensor operators.

Q. III-3) Can DomainTools tell what queries I'm issuing? What if I need assured query privacy as a matter of policy, regulation or operational security?

DomainTools logs all queries made to DNSDB for accounting- and troubleshooting-related purposes. If you need assured query privacy, DNSDB Export (which leverages an on-premises copy of DNSDB) will allow you to have that.

S. IV) Technical Jargon-Related Questions

Q. IV-1) What is the difference between "recursive resolvers" and "authoritative nameservers"

Recursive resolvers are used by users to resolve the names of the sites they're interacting with, whatever and wherever those might be. For example, if you visit www.cnn.com, a recursive

resolver will translate that domain name to the IP address your computer needs. Recursive resolvers are most commonly run by ISPs, enterprises, colleges or universities, etc., for the benefit of their local users, although some recursive resolvers may be intentionally open to anyone, like Google's 8.8.8.8.

Authoritative nameservers are different. They get designated by the domain owner when the domain owner registers a new domain name, and are used to describe the relationship between domain names and the IP addresses used by that specific domain. Authoritative nameservers may be run by the domain owner directly, or by a third party such as a domain name registrar or hosting company. Authoritative nameservers only know about/answer for the specific domain names assigned to them.

Q. IV-2) What is a "bailiwick?"

The bailiwick of a content DNS server is quite a simple notion. It is the domain that was used in the referral that directed a resolving proxy DNS server to that content DNS server in the first place. When a superdomain's content DNS servers issue a referral saying "Ask those servers over there about that particular domain.", then the domain in the referral is the bailiwick of the content DNS servers when they come to be queried.

For example, if the net. content DNS servers respond to an enquiry for the name an.example.net. with a referral to content DNS servers at 10.0.0.1 and 10.0.0.2 for the domain example.net., then the bailiwick of the latter servers, when they come to be queried, is example.net..

Bailiwick is the scope of authority of any particular content DNS server, determined by following a chain of referrals from the root of the DNS namespace. A content DNS server may only be trusted where the information it provides is about names within its own bailiwick.

Q. IV-3) What's a "base domain?"

A "base domain" is what registrants purchase from a registrar when they buy a new domain name. For example, nytimes.com is a base domain name.

Q. IV-4) What's a "fully qualified domain name?"

A "fully qualified domain name" is any hostname, and usually includes a base domain name. For example, www.cnn.com is a fully qualified domain name. printer23 is an example of a local domain name that is not fully qualified.

Q. IV-5) What's an "RRset?"

All DNS resource records of the same name, class, and type from a DNS response. For example, a server that is doing load balancing via DNS might have two, three, or even more A records for a given fully qualified domain name. (see RFC2136 just above section 1.1)

For example:

```
www.google.com.     300     IN      A       74.125.227.145
www.google.com.     300     IN      A       74.125.227.148
www.google.com.     300     IN      A       74.125.227.146
www.google.com.     300     IN      A       74.125.227.144
www.google.com.     300     IN      A       74.125.227.147
```

Q. IV-6) What are the different RRTypes and what do they show? Examples of each?

DNS Resource Record types are described at RFC6895 section 3.1.

Some DNS record types are very common, including (but not limited to):

Record Type	**Function**
A	Maps a domain name to IPv4 address
AAAA	Maps a domain name to IPv6 address
CNAME	Maps one domain name to another
NS	Defines a domain's nameserver
PTR	Maps an IP address to a domain name
MX	Defines a domain's mail exchanger
TXT	Returns some specified text content

Another DNS record type that's less-common is the SRV record. SRV records are defined in RFC 2782 from February 2000, co-authored by DomainTools' very own Dr. Paul Vixie.

SRV records allow a site to instantiate services on non-standard port numbers, and to easily load balance services across multiple servers of varying size. Some sites, including DomainTools, find this quite useful and rely heavily on SRV records.

Examples of the various record types include:

```
www.princeton.edu.      67  IN  A  140.180.223.42
mx.smtp.ucla.edu.       3600    IN  AAAA
2607:f010:3fe:102::ff:fe01:ac
```

```
www.uoregon.edu.        60  IN  CNAME   drupal-cluster5.uoregon.edu.
caltech.edu.            43200   IN  NS  tepid.ni.caltech.edu.
112.4.193.128.in-addr.arpa. 3600    IN  PTR www.orst.edu.
columbia.edu.           3600    IN  MX  10 mail-in.cc.columbia.edu.
ucdavis.edu.            14400   IN  TXT "v=spf1 ip4:198.17.84.4/32
    ip4:198.17.84.15/32 ip4:128.120.0.0/16 ip4:169.237.0.0/16
    ip4:152.79.0.0/16 include:stspg-customer.com include:sendgrid.net
    include:spf.boardbooks.com ~all"
```

See also The Magic of SRV Records.

Q. IV-7) What's "RRname" and what is its significance?

An RRname is a Resource Record Name or DNS label. This is the left hand side of a DNS record:

```
www.princeton.edu.      67  IN  A       140.180.223.42
```

Q. IV-8) What's "Rdata"?

Rdata is the value (or right hand side) of the DNS record.

```
www.princeton.edu.      67  IN  A       140.180.223.42
```

S. V) Usage Questions

Q. V-1) I'd like to get more than the 10,000 results in the web interface (or more than the 1,000,000 results in the API/CLI interface). What are my options?

You can do this if you have an API key subscription to access the data.

If you routinely need to do queries resulting in more than 1,000,000 results, the best path forward is DNSDB Export. When you have an on-premises copy of the database, you can tailor how you access the data and potentially retrieve an unlimited number of matching records.

Q. V-2) What output formats are available?

DNSDB Scout™, displays results in the browser window and is available for Google Chrome and Firefox, as well as in a web site most browsers can access.. You can download results as a JSON or CSV file, or copy and paste that data. See DNSDB Scout for more information

The web GUI interface at https://www.dnsdb.info/ is intended solely for occasional use, displays results in the browser window. You can copy and paste that data into a report or plain text file if you want to save those results.

The Python CLI dnsdb_query.py client offers more flexibility. Output formats for the dnsdb_query.py client include plain text (the default) and JSON format via the -j or --json options.

The C-language CLI dnsdbq client supports plain text, JSON, CSV and DNS formatted output, as well as querying for your remaining quota. It is open source on github at .

See also the “Example scripts” section of the DNSDB API documentation.

Q. V-3) Can you provide some examples of how I might write time-constrained DNSDB queries?

dnsdb_query.py –help states:

```
Usage: dnsdb_query.py [options]
```

Options:

```
 [...]
--before=BEFORE only output results seen before this time
--after=AFTER       only output results seen after this time
```

Time formats are: "%Y-%m-%d", "%Y-%m-%d %H:%M:%S",
"%d" (UNIX timestamp), "-%d" (Relative time in seconds),
BIND format (e.g. 1w1h, (w)eek, (d)ay, (h)our, (m)inute, (s)econd)

Some examples include:

Show entries from the last hour:

```
$ dnsdb_query.py -r www.google.com --after=1h
```

Show entries from after 2015-6-18:

```
$ dnsdb_query.py -i 216.170.114.3 --after=2015-6-18
```

Show entries from between the dates shown:

```
$ dnsdb_query.py -i 216.170.114.3 --after=2015-6-18 --before=2016-1-1
```

See also the “Lookup methods” section of the DNSDB API documentation which states:

You may filter results by time using the time_first_before, time_first_after, time_last_before, and time_last_after query parameters. These parameters expect a UTC timestamp with seconds granularity or a relative time in seconds (preceded by -).

Q. V-4) Can I do time-constrained (i.e. time fencing) queries in the web interfaces or DNSDB Scout™?

Yes, we support time-fencing in DNSDB Scout.

Q. V-5) What’s the difference between -r, -n, and -i queries in the DomainTools-provided dnsdb_query.py and dnsdbq DNSDB clients?

The different options are:

-r: queries the RRset
-n: queries the Rdata by name
-i: queries the Rdata by IP address or by CIDR prefix
For more, see RRset and Rdata Demystified.

If you are starting with an IP address, you must use -i because that’s the only option that allows you to query by IP address:

```
$ dnsdb_query.py -i 104.244.13.104
fsi.io. IN A 104.244.13.104
www.fsi.io. IN A 104.244.13.104
olddocs.fsi.io. IN A 104.244.13.104
farsighsecurity.com. IN A 104.244.13.104
www.farsighsecurity.com. IN A 104.244.13.104
farsightsecurity.com. IN A 104.244.13.104
www.farsightsecurity.com. IN A 104.244.13.104
```

```
archive.farsightsecurity.com. IN A 104.244.13.104
```

If you are starting with a domain name, you will normally want to use the -r option:

```
$ dnsdb_query.py -r www.fsi.io
;;  bailiwick: fsi.io.
;;      count: 57
;; first seen: 2013-10-08 21:20:49 -0000
;;  last seen: 2014-12-04 21:38:28 -0000
www.fsi.io. IN A 66.160.140.76

;;  bailiwick: fsi.io.
;;      count: 36
;; first seen: 2015-06-07 06:13:14 -0000
;;  last seen: 2016-03-09 02:40:20 -0000
www.fsi.io. IN A 104.244.13.104

;;  bailiwick: fsi.io.
;;      count: 5
;; first seen: 2013-12-19 14:36:40 -0000
;;  last seen: 2014-10-17 18:57:45 -0000
www.fsi.io. IN AAAA 2001:470:b0::76

;;  bailiwick: fsi.io.
;;      count: 11
;; first seen: 2015-06-09 10:30:06 -0000
;;  last seen: 2015-07-29 08:45:55 -0000
www.fsi.io. IN AAAA 2620:11c:f004::104
```

The most common time when you will use -n, is when you're searching for domains associated with a nameserver.

```
$ dnsdb_query.py -n ns7.dnsmadeeasy.com
3dg.biz. IN NS ns7.dnsmadeeasy.com.
chal.biz. IN NS ns7.dnsmadeeasy.com.
cpcl.biz. IN NS ns7.dnsmadeeasy.com.
g3ms.biz. IN NS ns7.dnsmadeeasy.com.
icti.biz. IN NS ns7.dnsmadeeasy.com.
u3o8.biz. IN NS ns7.dnsmadeeasy.com.
```

```
90501.biz. IN NS ns7.dnsmadeeasy.com.
cpynl.biz. IN NS ns7.dnsmadeeasy.com.
cvent.biz. IN NS ns7.dnsmadeeasy.com.
[etc]
```

The equivalent commands using the dnsdbq cli client to the above commands are:

```
$ dnsdbq -i 104.244.13.104
$ dnsdbq -r www.fsi.io
$ dnsdbq -n ns7.dnsmadeeasy.com
```

Q. V-6) How can I check my quota?

You can use a command such as:

```
$ dnsdbq -I
```

or

```
$ curl --header "X-API-Key: my-api-key-here"
https://api.dnsdb.info/lookup/rate_limit
```

Your API Key will normally be in .dnsdb-query.conf in your home directory.

For more information, please refer to Service Limits section of the DNSDB API documentation.

Q. V-7) Do you support mid-string wildcard searches, or searches using regular expressions?

Yes via DNSDB Flex.

Q. V-8) Do you support wildcards in the form of CIDR prefix notation or IP ranges?

Yes, using Flex, or when using the DomainTools supplied dnsdb_query.py client you can search by CIDR prefix or by IP range. Example of a permitted query:

As of July, 2022 DNSDB was changed to reduce the amount of junk wildcard domains in its database. We are gradually rolling out a change to replace multiple wildcarded DNS rrnames with a single rrname that starts with a _WILDCARD_. label. No other rrname labels contain

uppercase letters, so records with this (all upper case) _WILDCARD_. were never in DNSDB before. Note that there are existing, real, domain names that contain a _wildcard_. label (all lower case).

```
$ dnsdb_query.py -i 128.223.0.0/16
```

or equivalently

```
$ dnsdb_query.py -i 128.223.0.0-128.223.255.255
```

Some seemingly-equivalent-looking queries, however, will fail:

```
$ dnsdb_query.py -i 128.223.*
HTTP Error 400: Bad Request
$ dnsdb_query.py -i 128.223.*.*
HTTP Error 400: Bad Request
```

CIDR prefix queries are also supported for IPv6:

```
$ dnsdb_query.py -i 2001:48A8::/32
```

or equivalently

```
$ dnsdb_query.py -i
2001:48A8::-2001:48A8:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
```

The equivalent command using the dnsdbq CLI command is:

```
$ dnsdbq -i 2001:48A8::-2001:48A8:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF |
fgrep -v ';' | grep .
```

Q. V-9) How many parallel (or “simultaneous”) queries can I have outstanding simultaneously via the API?

Ten (10).

Q. V-10) I want to be able to guarantee that my queries to the database are not observed. Is there any way for me to obtain a complete copy of the database for use “on premises” within a secure enclave that’s not connected to the Internet?

DNSDB Export (an on-premises installation of DNSDB) provides total query privacy.

Q. V-11) I’d like to have real time access to the full stream of DNS data as it’s added to DNSDB. Is there some way to get that?

If you have a process that can utilize the real-time stream of observations that goes into DNSDB, DomainTools makes several different streams of data available to participants on the Security Information Exchange (SIE). The streams are created as byproducts of deduplication and filtering during processing before the data is put into the database.

To request a demonstration or to inquire about a trial API key please contact the DomainTools sales team.

Q. V-12) I’d like to be able to routinely watch for just select terms of interest (keywords, brands, etc.). Do you have some way to support that?

Yes, you can do this using SIE’s Newly Observed Domains and Newly Observed Hostnames. To request a demonstration or to inquire about a trial API key please contact the DomainTools sales team.

Q. V-13) How do I search for subdomains in DNSDB? In CLI/API and WebUI?

Please see Lookup methods section of the DNSDB API documentation.

As an example of searching for all domains in cs.uoregon.edu, you’d enter:

```
$ dnsdb_query.py -r \*.cs.uoregon.edu
```

When using the CLI, shell escape the- by preceding it with a backslash. When using the WebUI, omit the backslash.

Q. V-14) For some reverse-IP lookups, there's lots of results, often associated with cheap web hosts experiencing high client turnover. Is there a way to limit the results to only current results? Or, at a minimum, only receive the result from the most recent date that the dns name resolved to that IP.

You can use time fencing to limit the results returned. See above in question VI-3.

For example, to get results from just the last hour:

```
$ dnsdb_query.py -r www.google.com --after=1h
```

You can also sort results by time last seen:

```
$ dnsdb_query.py -s time_last -r www.google.com
```

Q. V-15) Is it possible to find DNS responses from only a particular nameserver IP?

While this can be done by monitoring Channel 202 in Security Information Exchange (SIE), it is not currently possible in DNSDB.

Q. V-16) I forgot my password? How do I reset my password?

Request a password reset by contacting EnterpriseSupport@domaintools.com.

Q. V-17) Do you "lock" accounts?

Generally, we only lock accounts if we see evidence of compromise, use inconsistent with DomainTools' terms of service, or we are unable to reach a user at their email address of record. If you believe your account may have been locked, please contact EnterpriseSupport@domaintools.com for assistance.

Q. V-18) How do I request credentials for a new teammate in my organization, or what if a team mate has left and I'd like to retire their credentials?

Your registered point of contact for your contract should contact EnterpriseSupport@domaintools.com.

Q. V-19) How do I learn about maintenance notifications?

Contact EnterpriseSupport@domaintools.com and ask to be added to the notification list.