DNS lab: dig, part 1

In the following labs, we'll be using "auth1" as your work machine.
In reality, this is not very important, as we're only going to be using
the 'dig' command.

DIG
---

1. Issue DNS queries using 'dig':

Note: make sure you explicitly specify the nameserver to query using the
"@" syntax in dig:

```
    $ dig @server_ip ...
```

If you do not specify the @server_ip, then dig will use the nameserver(s)
listed in /etc/resolv.conf

1a. Run each command, look for the ANSWER section and write down the result.
    Make a note the TTL as well.

Repeat the command. Is the TTL the same? Are the responses Authoritative?

```
                                                      RESULT 1        RESULT 2
                                                      --------        --------
    $ dig @10.20.0.254  your-favorite-domain a
    $ dig @10.20.0.254  www.google.com. a
    $ dig @10.20.0.254  afnog.org. mx
    $ dig @10.20.0.254  NonExistentDomain.sometld any
    $ dig @10.20.0.254  tiscali.co.uk. txt
    $ dig @10.20.0.254  www.afrinic.net aaaa
    $ dig @10.20.0.254  ipv6.google.com aaaa
```

1b. Now send some queries to another caching server.

    (Run each of the following twice, and note the time in ms for each attempt)

```
                                                      RESULT 1        RESULT 2
                                                      --------        --------
    $ dig @8.8.8.8 news.bbc.co.uk. a
    $ dig @208.67.222.222 yahoo.com. a
    $ dig @<a server of your choice> <domain of your choice> a
```

    How long did it take each answer to be received? (on the first, and
    on the second lookup)

2. Reverse DNS lookups

    Now try some reverse DNS lookups - note here that we do not explicitly
      specify which nameserver dig should query. Which nameserver will be used ?

```
    $ dig -x 10.20.X.1
    $ dig -x 10.20.X.2
    $ dig -x 10.20.X.3
```

    ... where X is in the range 1-25

    Repeat for an IP address of your choice, on the Internet. Remember, you'll have to use @10.20.0.254 to be able to perform DNS queries on the Internet...

Now try to lookup:

```
$ dig 1.X.20.10.in-addr.arpa. PTR
```

... where X is in the range 1-25.

What do you notice ?

Let's try IPv6 now:

```
$ dig -x 2001:42d0::200:2:1
```

What are the differences you can observe in the results, between reverse DNS for IPv6 and IPv4 addresses ?

Note: you may possibly not get an answer for the v6 address - but compare the question section for the IPv4 and IPv6 reverse addresses.

3. DNSSEC & EDNS0

Try some of the queries above, this time add the "+edns=0" option.

For example:

```
$ dig @10.20.0.254 www.icann.org +edns=0
```

(you may want to use "more" to limit the output of the command to one screen at a time)

```
$ dig @10.20.0.254 www.icann.org +edns=0 | more
```

Notice the OPT PSEUDOSECTION, at the top of the output ?

What do you notice about the flags: section in the OPT section ?

Let's explicitly enable the BUFSIZE option, but not EDNS0:

```
$ dig @10.20.0.254 www.icann.org +bufsize=1024 | more
```

Notice that EDNS is set automatically, and notice the udp: size section in the OPT pseudosection.

Now, let's try and retrieve DNSSEC records:

```
$ dig @10.20.0.254 isoc.org DNSKEY | more
$ dig @10.20.0.254 www.isoc.org RRSIG | more
```

And finally, let's tell our DNS server that we support DNSSEC:

```
$ dig @10.20.0.254 www.isoc.org A +dnssec
$ dig @10.20.0.254 isoc.org NS +dnssec
```

Do you notice a new field in the "flags:" section of the answer ?

```
$ dig @10.20.0.254 www.isoc.org A
$ dig @10.20.0.254 isoc.org NS
```

Compare with doing dig WITHOUT the +dnssec option:

If you are already running a nameserver in your group,
what happens if you send DNSSEC enabled queries to it ?

```
$ dig @10.20.XXX.3 noc.dns.nsrc.org A +dnssec
$ dig @10.20.XXX.3 dns.nsrc.org NS +dnssec
```

... where XXX is the number of your group, and .3 is your caching server (maybe it isn't set up yet!)