DNS/DNSSEC Workshop in conjunction with Lanka Network Operators Group (LKNOG) Conference 12-16 August 2024

Lab Exercises

Champika Wijayatunga
Technical Engagement Sr. Manager - Asia Pacific
<champika.wijayatunga@icann.org>

Introduction to dig

Working with "dig" and understanding its outputs are crucial for DNS troubleshooting and debugging, so don't be shy to ask questions.

Note: We can't explore all dig possibilities during this workshop. We do recommend you to continue playing and learning from this fantastic tool.

Goals

- Become familiar with the use of dig
- * Use dig to retrieve and examine DNS ressource records
- Interpret dig's output and identify some common DNS problems.

> Notes
> Commands preceded with "$" imply that you should execute the command as a general user - not as root.
> Commands preceded with "#" imply that you should be working as root.
> Commands with more specific command lines (e.g. “rtrX>"or “mysql>") imply that you

> are executing commands on remote equipment, or within another program.

The dig Tool

The tool "*dig*" was originally shipped with BIND and is commonly found on many Unix-like platforms. It stands for *Domain Information Groper*. It collects data about Domain Name Servers and is helpful for troubleshooting DNS problems and/or displaying DNS information.

Other DNS implementations also include similar tools, often with similar names (e.g. kdig). Older tools used for DNS troubleshooting include *nslookup* and *host*.

A manual page for dig can be found here or from the command-line. There are a lot of available parameters. You can ignore most of them while you are getting started.

```
$ man dig
```

A typical invocation of dig looks like:

```
dig @SERVER NAME TYPE
```

> SERVER: name or IP address of the name server to query. If no server argument is provided, dig consults /etc/resolv.conf. If no usable addresses are found, dig will send the query to the local host.
> NAME: name of the resource record that is to be looked up.
> TYPE: type of query requested (A, MX, NS, SIG, ...). If no type supplied, dig will lookup for an A record.

For each dig query sent, a response is expected on the terminal with different sections and lines. Here are few of them:

- The first line displays the version of the dig command.
- The **HEADER** section shows the information it received from the server. **Flags** refer to the answer format and they are extremely important to understand the overall result of the query. There are six (06) flags:

    1. AA: Authoritative Answer
    2. TC: Truncated Response
    3. RD: Recursion Desired
    4. RA: Recursion Available
    5. AD: Authentic Data
    6. CD: Checking Disabled

- The OPT PSEUDOSECTION displays advanced data such as EDNS (Extension

mechanisms for DNS), if used.

- The QUESTION section displays the query data that was sent.
- The ANSWER section: probably the most important section for the user.
- The STATISTICS section shows metadata about the query: Query time 9amount of time it took to get the response); SERVER (IP address and port of the responding DNS server); WHEN (timestamp when the command was run); MSG SIZE rcvd (size of the reply from the DNS server).

Sending DNS Queries Using dig

Try using dig to look up the address corresponding to the DNS name www.icann.org. Here are various ways of doing that; what differences do you see in the output from each of them?

```
$ dig www.icann.org A
$ dig www.icann.org A
$ dig @8.8.8.8 www.icann.org A
$ dig @1.1.1.1 www.icann.org A
```

For each answer that you got, discuss the different section of the answer with the facilitators. You will be surprised to see that dig tool provides a sea of information.

Try now other record type

```
$ dig NS icann.org
```

```
$ dig SOA ricta.org.rw
```

```
$ dig www.ricta.org.rw A
```

```
$ dig @8.8.8.8 www.ricta.org.rw A
```

```
$ dig SOA ricta.org.rw @ns1.ricta.org.rw.
```

Again, try to discuss the various outputs with your instructors.

Let's continue exploring dig tool with the following examples

- +short: to display only the queried resource record value

```
$ dig @8.8.8.8 www.icann.org A +short
```

- +noall +answer: to get detailed information of the answers section only

```
$ dig @8.8.8.8 www.icann.org A +noall +answer
```

- +trace: lists each different server the query goes through to its final destination. Good for troubleshooting

```
$ dig www.icann.org A +trace
```

- -x: to lookup a domain name by its IP address: **reverse lookup**

```
$ dig -x 192.0.47.7
```

- -f: to look up multiple entries stored in a file

```
$ echo "icann.org google.com gmail.com" > test_batch_lookup.txt ; dig -f
```

- hostname.bind: to retrieve the hostname of the server (if allowed to)

```
$ dig CHAOS txt hostname.bind @ns.icann.org.
$ dig txt hostname.bind @d.root-servers.net CHAOS
```

- id.server:

```
$ dig txt ID.SERVER @d.root-servers.net CHAOS
$ dig txt ID.SERVER @9.9.9.9 CHAOS
```

- version.bind: BIND servers respond to queries for name version.bind with record type TXT and class CHAOS. By default, this is set to the version of BIND that has been installed

```
$ dig CHAOS txt version.bind @ns.icann.org.
$ dig txt version.bind @d.root-servers.net CHAOS
```

Talking with DNSSEC using dig

- Getting the resource record signatures:

```
$ dig www.icann.org A +dnssec
```

```
$ dig icann.org NS +dnssec
```

Compare the number of RRSIG you get in the first case to the number you received in the second case.

You can add the `+multi` option to make the results more "readable".

- Retrieve the public keys for the zone: they are stored in a specific resource record type named "DNSKEY"

```
$ dig icann.org DNSKEY
```

You can mix the known options such as redirecting to a specific name server, adding multilign option, etc.

- Retrieve the delegation signer info for the zone: they are stored in a specific resource record type named "DS"

```
$ dig icann.org DS
```