A dry run (or a practice run\footnote{1}) is a testing process where the effects of a possible failure are intentionally mitigated. \cite{Wikipedia}
DNSSEC deployment is lagging

- Top Level Domains are 91.6% signed, but Second Level Domains are certainly not!

<table>
<thead>
<tr>
<th>TLD</th>
<th># delegations</th>
<th># signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>com</td>
<td>158.6M</td>
<td>4657852 (2.94%)</td>
</tr>
<tr>
<td>tk</td>
<td>24.7M</td>
<td>0</td>
</tr>
<tr>
<td>de</td>
<td>17.0M</td>
<td>257777 (1.52%)</td>
</tr>
<tr>
<td>cn</td>
<td>15.3M</td>
<td>1460 (0.01%)</td>
</tr>
<tr>
<td>net</td>
<td>13.5M</td>
<td>442813 (3.28%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TLD</th>
<th># delegations</th>
<th># signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>uk</td>
<td>11.0M</td>
<td>237762 (2.16%)</td>
</tr>
<tr>
<td>org</td>
<td>10.4M</td>
<td>412124 (3.96%)</td>
</tr>
<tr>
<td>nl</td>
<td>6.2M</td>
<td>3581652 (57.77%)</td>
</tr>
<tr>
<td>ru</td>
<td>5.7M</td>
<td>5602 (0.10%)</td>
</tr>
<tr>
<td>br</td>
<td>4.9M</td>
<td>819821 (16.73%)</td>
</tr>
</tbody>
</table>

Combined results from Verisign Domain Name Industry Brief and DNSSEC and DANE Deployment Statistics
DNSSEC deployment

- Top Level Domains are 91.6% signed, but Second Level Domains are not!
- What is the benefit?

Graph from "Internet Survey of DANE/TLSA DNS Records: Application and Use for Mail Exchanger" presented at OARC 37 by Dr Erwin Hoffmann
DNSSEC deployment is lagging

- Top Level Domains are 91.6% signed, but Second Level Domains are certainly not!
- What is the benefit?
- DNSSEC failure appears as Network Failure
DNSSEC deployment is lagging

- Top Level Domains are 91.6% signed, but Second Level Domains are certainly not! Why?
- What is the benefit?
- DNSSEC failure appears as Network Failure
- Enabling DNSSEC is not without risks
  - Unforeseen failures
  - Once the DSes are published, it takes TTL time to revert!
DNSSEC deployment is lagging

- Efforts to ease DNSSEC deployments
  Increase confidence for operators to deploy:
  - Extended DNS Errors - [RFC8914]
    - DNSSEC failure not appearing as network failure
DNSSEC deployment is lagging

● Efforts to ease DNSSEC deployments
  □ Increase confidence for operators to deploy:
    ○ Extended DNS Errors - RFC8914

Failure not appearing as network failure

```
willem@makaak:~$ dig @1.1.1.1 www.dnssec-failed.org
;; <<< DiG 9.16.15-Ubuntu >>> @1.1.1.1 www.dnssec-failed.org
;; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 9428
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; EDE: 9 (DNSKEY Missing): (no SEP matching the DS found for dnssec-failed.org.)
;; QUESTION SECTION:
;www.dnssec-failed.org. IN A

;; Query time: 163 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: wo mrt 02 16:09:13 CET 2022
;; MSG SIZE  rcvd: 107
```
DNSSEC deployment is lagging

Efforts to ease DNSSEC deployments increase confidence for operators to deploy:

- Extended DNS Errors - RFC8914

DNSSEC failure not appearing as network failure

Hmm. We’re having trouble finding that site.

We can’t connect to the server at www.dnssec-failed.org.

If that address is correct, here are three other things you can try:

- Try again later.
- Check your network connection.
- If you are connected but behind a firewall, check that Firefox has permission to access the Web.
DNSSEC deployment is lagging

- Efforts to ease DNSSEC deployments
  Increase confidence for operators to deploy:
    - Extended DNS Errors - RFC8914
      - DNSSEC failure not appearing as network failure
    - Extended DNS Error reporting - draft-ietf-dnsop-dns-error-reporting
      - Notify operators about errors in their deployments
DNSSEC deployment is lagging

- Extended DNS Error reporting - draft-ietf-dnsop-dns-error-reporting
  - Reports after the fact!
DNSSEC deployment is lagging

- Extended DNS Error reporting - draft-ietf-dnsop-dns-error-reporting
  - Reports after the fact!
- Our idea: **dry-run DNSSEC**
  - Get the DNS Error reports, without the end-user experiencing the failure
  - Try out a DNSSEC deployment live for a period of time, before rolling to actual DNSSEC
Dry-run DNSSEC - signalling

- Signalled with a new Delegation Signer (DS) Algorithm

[Image of DNSSEC Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms]

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Status</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>-</td>
<td>[RFC3658]</td>
</tr>
<tr>
<td>1</td>
<td>SHA-1</td>
<td>MANDATORY</td>
<td>[RFC3658]</td>
</tr>
<tr>
<td>2</td>
<td>SHA-256</td>
<td>MANDATORY</td>
<td>[RFC4509]</td>
</tr>
<tr>
<td>3</td>
<td>GOST R 34.11-94</td>
<td>OPTIONAL</td>
<td>[RFC5933]</td>
</tr>
<tr>
<td>4</td>
<td>SHA-384</td>
<td>OPTIONAL</td>
<td>[RFC6605]</td>
</tr>
<tr>
<td>TBD</td>
<td>DRY-RUN</td>
<td>OPTIONAL</td>
<td>[draft-yorgos-dnsop-dry-run-dnssec]</td>
</tr>
<tr>
<td>5-255</td>
<td>Unassigned</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Dry-run DNSSEC - signalling

- Signalled with a new **Delegation Signer (DS) Algorithm**
- First byte of the *Digest* field used as *Digest Type*
- Remainder as *Digest* for that type

```
  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+--------+-----------------+---------+--------------------------+
| Key Tag | Algorithm        | DRY-RUN |
+--------+-----------------+---------+--------------------------+
| Digest Type |                  / |
+--------+-----------------+---------+--------------------------+

```
Dry-run DNSSEC - signalling

```
+-----------------+-----------------+-------------------+
| Key Tag          | Algorithm        | DRY-RUN           |
+-----------------+-----------------+-------------------+
| Digest Type     | Digest           |                   |
+-----------------+-----------------+-------------------+
```

- Example:
  - `ietf.org. DS 45586 5 1 D0FDF996D1AF2CCDBDC942B02CB02D379629E20B`
    as dry-run is:
  - `ietf.org. DS 45586 5 DRY-RUN 01D0FDF996D1AF2CCDBDC942B02CB02D379629E20B`
Dry-run DNSSEC - operation

1. Validate RRset
2. DSes?
   - yes: Select DS
     - Prefer the DRY-RUN DS
     - DRY-RUN DS?
       - yes: Remove DRY-RUN DSes From DS RRset
       - no: DNS Error reporting
   - no: DNSSEC valid?
     - yes: DNSSEC valid? (set AD bit)
     - no: INSECURE
3. BOGUS SERVAIL
   - no: DNSSEC valid?
     - yes: SECURE
     - no: no
Dry-run DNSSEC operation

- Validate RRset
- DSES?
  - yes
    - Select DS
      - Prefer the DRY-RUN DS
      - Remove DRY-RUN DSes From DS RRset
  - no
- DRY-RUN DS?
  - yes
  - DNSSEC valid?
    - yes
    - Secure set AD bit
    - Opportunistic DANE
  - no
- DNSSEC valid?
  - yes
  - Secure set AD bit
  - Opportunistic DANE
- INSECURE
  - no
- BOGUS
  - no
  - DNSSEC valid?
    - no
    - INSECURE
    - yes
    - SECURE
      - set AD bit
Dry-run DNSSEC - backwards compatibility

- RFC6840 Section 5.2

"... when determining the security status of a zone, a validator disregards any authenticated DS records that specify unknown or unsupported DNSKEY algorithms or unknown or unsupported message digest algorithms. If none are left, the zone is treated as if it were unsigned."

slightly modified to make the text more concise

- Dry-run DNSSEC is like a DS with an unknown Digest-type which is just like a missing DS - but with DNS Error Reporting
RIPE Atlas measurements

- Measure backwards compatibility by Tom Carpay during hackathon
CDNSKEY & CDS Considerations

- CDS works as before. _But watch out with CDNSKEY!_

- To introduce a new dry-run DS:
  - CDS only, or
  - CDS MUST be present alongside a CDNSKEY

  dnsprivacy.org. CDS 49150 13 DRY-RUN 010e6083a30e9758e49eab4c135602b4a71a87533a

  dnsprivacy.org. CDNSKEY 257 3 13 (55/VHWP2dPtUN1GIMwn1o3UKUKyo65wgNPykTg3i0AU4
  I14MgLTher7MtYn4RIZFQ4Cuk7ozhqQ1HHe+Za5w== )
  ) ;{id = 49150 (ksk), size = 256b}

- Real DNSSEC can be bootstrapped from dry-run DNSSEC
Wet-run query flag

- Clients can opt-in to dry run dry-run DNSSEC
- Resolver should process queries with the flag as if DRY-RUN DSes are real

```
+0 (MSB) +1 (LSB)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0: |     EXTENDED-RCODE    |           VERSION            |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2: |  DO| DD|       Z       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```
Wet-run flag disguised as EDNS0 option

- Send along with dry run dry-run (i.e. wet-run) queries:

```
0                       8                      16
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          Wet-run OPTION-CODE (TBD)            |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                 OPTION-LENGTH (0)             |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- Both flag and option cannot be tested (with RIPE Atlas) but the option is probably better supported...
Why not do * instead?

- DNS Error Reporting on signed zones without DS
  - Operators signal DNS Error Reporting support anyway
  - No stub level dry-run validation (stubs don’t get the signal)
  - No testing of (and handling of) the DS RRset
  - No Opportunistic DANE and no CDS handling
  - Not explicit what needs to be dry-run precisely

- Short (or control over) TTL of DS in parent
  - No long-term dry-run for a variety of scenarios
Dry-run DNSSEC
Questions, comments, suggestions ...