Public Resolver Operators' Implementation Guidelines

Note: This covers both Closed+Public and Open+Public resolver operators.

Most public resolver operators develop and maintain their own DNS server implementations (with some exceptions). As a result, we chose not to provide vendor-specific implementation guidelines that likely don't apply to them.

DNS Security and Privacy

1. **Practice 1:** DNSSEC validation **MUST** be enabled for recursive resolvers.

   To test that your recursive DNS resolver is actually performing validation, you can try the following:

   ```
   $ dig @ip.of.your.validator www.icann.org. SOA
   ```

   Check if the 'ad' bit is set in the 'flags' section of the response. If it is, then your resolver is performing DNSSEC validation.

2. **Practice 2 (Privacy Consideration):** QNAME minimization **MUST** be enabled to mitigate leakage of domain names.

   QNAME minimization is normally the default in modern resolver software; however, be sure to check that it is enabled.

3. **Practice 3 (Privacy Consideration)**: DoT (DNS-over-TLS) or DoH (DNS-over-HTTPS) **SHOULD** be enabled.

   Deploying either is the easiest way to protect against eavesdropping and manipulation of DNS queries and man-in-the-middle attacks by encrypting DNS queries between stub and recursive resolvers, or between a forwarding and recursive resolver.

DNS Availability and Resilience

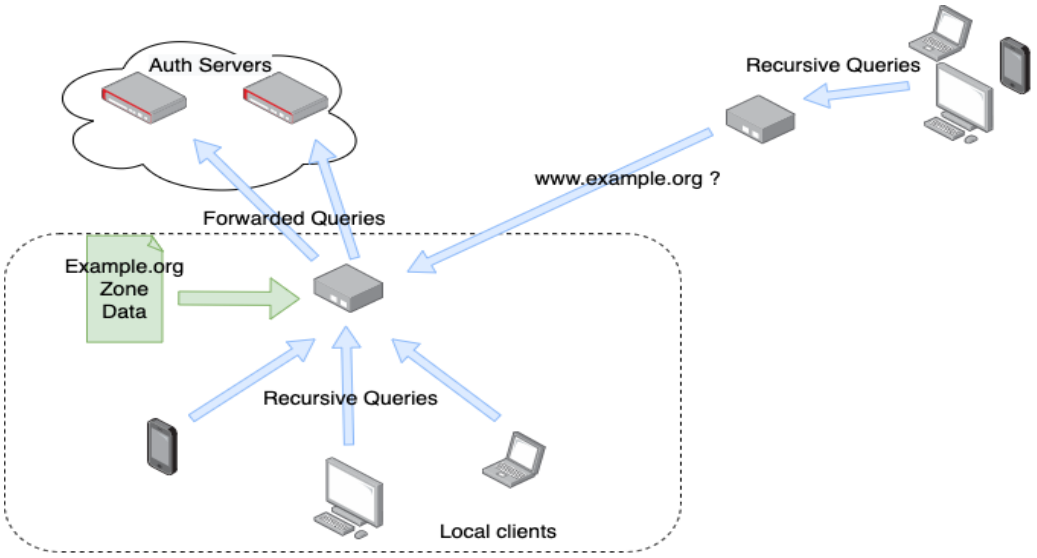
4. **Practice 4**: Authoritative and recursive DNS service **MUST NOT** coexist on the same DNS server.

   General Implementation Considerations:

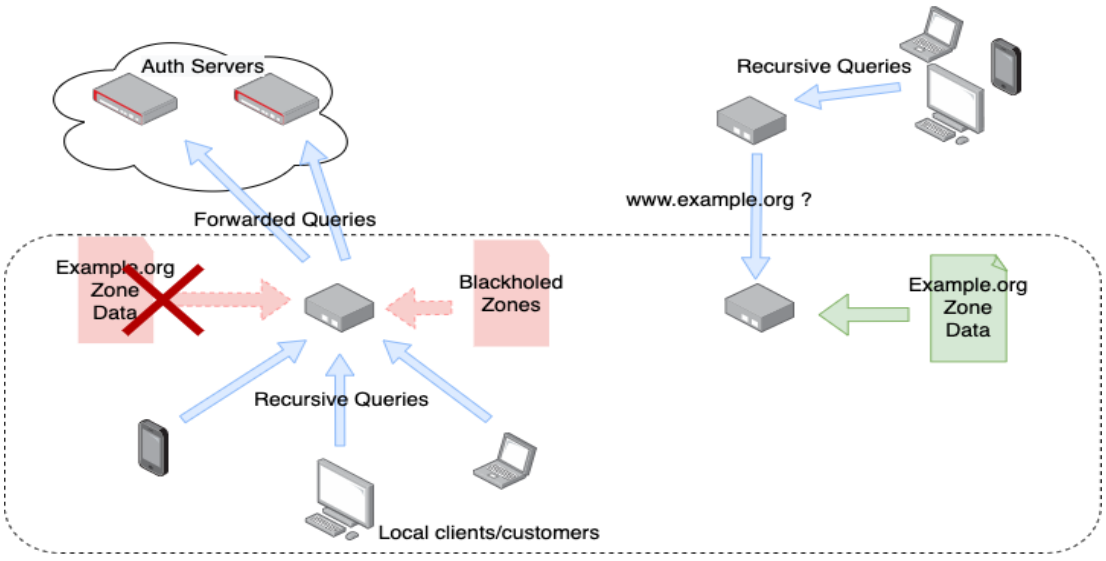
   Dedicated DNS recursive resolvers **should** be set up separately from the authoritative nameservers. Ideally, the recursive name servers will not be reachable from the wider

Internet (see the corresponding BCPs on Network and Service security using ACLs).

In the context of recursive servers, this means you should not configure them to also serve public authoritative zones even if the software allows it. An exception can be made for zones used to blackhole queries that should not be forwarded to the internet at large (such as RFC 1918 reverse lookup zones such as those managed by the AS112 project - see https://www.as112.net/). The diagram below illustrates how a server configured to run as both authoritative and recursive is reconfigured, with the authoritative DNS server being split out as a distinct service.



Nameserver acting as both Recursive and Authoritative



Separation of duties - dedicated Recursive resolver and Authoritative servers

5. **Practice 5:** Data collected through passive logging of DNS queries **MUST** only be retained for as long as is necessary for the sound operation of the service offered, including troubleshooting, research, and satisfying local legal requirements on data retention.

   General Implementation Considerations:

   There are no specific implementation guidelines for this requirement - individual business and legal requirements dictate how it will be implemented.

6. **Practice 6:** Your recursion services **MUST** have resilience by using at least two distinct servers that take diversity into consideration.

   Public resolver IP addresses are either handed out to clients using DHCP or another provisioning mechanism, or they are manually configured by end users who willingly choose to use a different resolver service than the one provided to them by their ISP or institution (if allowed).

   It is worth noting that all known public resolver operators offer at least two IP addresses in each protocol family (IPv4, IPv6), usually announced out of different Autonomous Systems. This is to avoid failures if one prefix became unavailable due to a routing misconfiguration or similar outage.

7. **Practice 7:** Monitoring of the services, servers, and network equipment that make up your DNS infrastructure **MUST** be implemented.

   General Implementation Considerations:

   Examples of resources and services that should be monitored:

   - Availability: does the DNS service answer?
     - Example: On port 53 UDP and TCP, does the DNS server return data if queried?
   - Correctness: does the DNS service return the expected data ?
     - Example: Query for a name and resource record type (for instance: www.example.org A, or example.org SOA), then check the result against a known good value
   - Latency: does the service respond in a timely fashion ?
     - Example: How long does it take for the service to respond to the above checks ? It should be within a reasonable timeframe, say less than 5ms for most authoritative queries, and probably less than 200-300ms for recursive queries. Account for the time it takes to fetch an answer if not already in the cache, and the network latency (round trip) between the monitoring service and the DNS service you are testing.

The above three tests can be performed as a single check using most monitoring platforms / services.

Example with the Nagios plugin, check_dig:

```
check_dig -4 -H a.icann-servers.net -l www.icann.org -w 2 -c 5 -a \
    www.vip.icann.org
```

The queried server is a.icann-servers.net, using IPv4 ('-4'). The queried name is 'www.icann.org'. The settings -w 2 and -c 5 set a warning and critical threshold if the server hasn't responded before 2 and 5 seconds have elapsed, respectively. `-a` is the expected result, in this case 'www.vip.icann.org'. We don't set a queried record type (-T), so it defaults to A.

Example output:

```
DNS OK - 0.123 seconds response time (www.icann.org.  3600 IN CNAME
www.vip.icann.org.)|time=0.122713s;2.000000;5.000000;0.000000
```

The same techniques apply for monitoring the availability and reachability of intermediate network devices - ICMP checks are often enough in most cases to detect failure of a router or switch in front of a DNS server.

If monitoring with a third-party service, there are many online providers which provide remote monitoring of availability and reachability (providing the service is using public IP addresses and filtering allows for remote monitoring).