The Split Horizon DNS Problem

ADD/DPRIVE/DNSOP Interim, Jan 2022
Ben Schwartz
Based on work by Tiru Reddy, Dan Wing, and Kevin Smith
1. Definitions
What is “split horizon DNS”?

- The network-provided resolver gives answers that are *meaningfully different* from the answers you will get from an external resolver.

- **Examples**
  - **Intranet domains**: `foo.corp.example.com` is NXDOMAIN in the public DNS.
  - **Internal variations**: `example.com` resolves to a canary instance when using Example Inc’s resolver.
  - **Network proximity**: `downloads.example.edu` resolves to a local server at each campus when using the resolver provided by that campus’ network.

- **Typical users**
  - Enterprise
  - Education
  - Home (power users, home gateways)
  - ISP (subscriber configuration panel)
What is “DNS hijacking”?  

- Answering queries for a domain name **without authorization by the zone**.
- DNS hijacking is not allowed within IETF standards:
  - RFC 2826: there must be “a single owner or maintainer to every domain … who is responsible for ensuring that each sub-domain of that domain has the proper records”.
  - DNSSEC regards hijacking of signed zones as Bogus.
- Examples
  - Answering for **www.example.com** without permission from the **example.com** zone.
  - Answering for **nonexistent.example.com** without permission (“**NXDOMAIN hijacking**”).
  - Inventing new Pseudo-TLDs without IANA/ICANN permission (i.e. NXDOMAIN hijacking at the root).

“Split-horizon DNS” - “DNS hijacking” = **“authorized split-horizon DNS”**.
2. Intractable Topics
Trust relationships

- If a device and network are not managed by the same entity, they might not have a high degree of mutual trust. In this, each party will choose its own DNS behavior (such as which resolvers to use) independently.
  - Likely driven by each party’s security assumptions.
- ADD Charter: “Making any recommendations about specific policies for clients or servers is out of scope.”
Unachievable goals for standards development

- Make clients and networks work together if they have incompatible security assumptions.
- Make people change their security assumptions.

Plea: Let’s focus on use cases where the only thing missing is a technical solution.
Looking for “interesting”

- We can still consider the effects of different policies, and define solutions that only work under certain policies.
- Clients that have a policy to *always* use the network-provided resolver will *always* see the split horizon names.
- Clients that have a policy to *never* use the network-provided resolver will *never* see the split horizon names.
- These cases are both *boring*.
- **What about clients that *sometimes* use the network-provided resolver?**
3. Hybrid Resolvers

A brief digression with puzzle pieces
**Hybrid Resolvers**

A *hybrid resolver* implements multiple resolution behaviors, and dispatches each question to an appropriate behavior according to a local policy.
Example: POSIX stub resolver with /etc/hosts
Example: “Authoritative Resolver”
The Interesting Case

- Both resolvers are hybrids,
- the network resolver has some local answers, and
- the client wants access to this resolver’s local zones.
- The other branches might be “Remote Answer”, “Iterative Resolution”, etc.
4. Scoping and Mechanisms
Achievable goals for standards development

**Easy**: Enable clients to learn about split-horizon names available on this network.

**Hard**: Enable clients to distinguish *DNS hijacking* from *authorized split horizon DNS*.
Announcing split-horizon names (the easy part)

- **Provisioning Domains (PvD)’s “dnsZones” extension (RFC 8801, Section 4.3)**
  - No authentication. The network operator can put anything they want in this extension!
  - PvD source is identified but not necessarily trusted.

- **Split DNS Configuration for IKEv2 (RFC 8598)**
  - No authentication, but the client usually has a close relationship with the tunnel operator.
  - Supports overriding DNSSEC trust anchors! (but recommends a user confirmation step)

- **DHCP Search Option (RFC 3397)?**
  - Not exactly the right semantics but really very similar to the PvD support.
Confirming authorization (the hard part)

- To confirm that a local resolver \(R\) is authorized to serve a given DNS zone, the client would need
  - (1) an identity for \(R\) (presumably an Authentication Domain Name (ADN))
  - (2) an assertion by the zone owner that it authorizes ADN to serve this zone
  - (3) a way to ensure that the assertion was not forged by \(R\)
  - (4) a secure transport to \(R\), authenticated to this ADN

- Example solutions (from draft-reddy-enterprise-dns-08)
  - DNR (draft-ietf-add-dnr) provides the ADN (1) and bootstraps the secure transport (4)
  - An NS record asserts that this ADN is authorized to serve the zone (2)
  - To prevent forgery (3), the client resolves the NS record
    - through an independent resolver over a secure transport, or
    - using local DNSSEC validation (if the claimed zone is signed and the client does local DNSSEC)
New DNS security standards have created new opportunities to make Split-Horizon DNS more secure and compatible.

We can make progress, but only if we focus on the use cases where a solution is possible.

Even a limited or inconvenient solution would be better than the status quo.