“What’s in a Name (collision)?”
Modeling and Quantifying Collision Potential
Casey Deccio, Verisign Labs
Workshop and Prize on Root Causes and Mitigation of Name Collisions (WPNC)
London, UK, March 10, 2014
Objectives

• Formalize a model of name resolution, based on current resolver library implementation.
• Define name collisions based on name resolution model.
• Define metrics to quantify probability and risk associated with name collision.
• Supply framework to apply model to network environments.
Motivation/Previous Work

• Previous work
  • “Name Collisions in the DNS”, Interisle Consulting Group (commissioned by ICANN)
  • “New gTLD Security, Stability, Resiliency Update: Exploratory Consumer Impact Analysis”, Verisign Labs

• “Outside in” perspective
  • Is data representative of current incidence and risk?
  • Can the risk be over- or under-estimated with outside data?

• New - “Inside out” perspective
  • What does data look like on the inside?
  • What is the risk potential?
Creating a Model of Name Resolution

**Benefits**
- Creates consistent reference.
- Facilitates definition of resolution behavior, such as name collision.
- Facilitates definition of metrics for quantification.
- Naturally leads to implementation.

**Requirements**
- Understand behaviors of resolver implementations.
- Model individual components and their dynamics.
- Represent as simply as is possible, and no more.
Resolver Library Behavior

- Suffix search list processing varies:
  - Across OS.
  - Depending on whether name is single- or multi-label.

- A series of names are queried to the DNS:
  - In specified order, built from search list.
  - Until positive response is returned from the DNS or list is exhausted.

Example

Name: “www”
Suffix search list: [“foo.example”]
Behavior: Windows XP; Query list: [“www.foo.example”]
Behavior: Linux; Query list: [“www.foo.example”, “www”]
Modeling Resolver Library Behavior

\[ Q_{(c,S)}(n) = [n_1, n_2, \ldots, n_m] \]

Sequence of DNS names recursively queried to produce the intended answer for \( n \).

- \( n_1 \ldots n_{m-1} \) result in negative responses.
- \( n_m \) produces the end result (positive or negative).
Modeling Mobility

• **“Home” environment** – network environment corresponding to resolver configuration
  - Suffix search list ($S$).
  - Locally administered DNS namespaces ($LA$).
• **Locality** – percentage of time clients operate within “home” environment ($h$).

“Private” (home-based) queries, answered by local authorities or public DNS

“Public” (away from home) queries, answered only by public DNS.
Name Collision – A definition

A name collision results from one or both of the following conditions:

A DNS query for any name, \( n_i \in [n_1, n_2, \ldots, n_{m-j}] \) \( j \in [0, 1] \) yields a positive result.

(\( j \) is 0 if \( n \) should return negative response; otherwise \( j \) is 1).

There is at least one name, \( n_i \in Q_{(c,S)}(n) \), such that the namespace for \( n_i \) is locally administered but the response to a DNS query for \( n_i \) is received the responded by by the public DNS.

Passive name collision – no difference in ultimate response.
Active name collision – difference in ultimate response.
Collision Probability

- **localAnswerFromPublic**
  - $1 - h$: if there is a locally administered name in query list, $Q_{(c,s)}(n)$
  - $0$: otherwise

- **falsePosProb $\leftarrow 0$**
  - If name in query list yields positive response from “public” query, then add $h$ to **falsePosProb**.
  - If name in query list yields positive response from “private” query, then add $1 - h$ to **falsePosProb**.

- **collisionProbability $\leftarrow \max(\text{localAnswerFromPublic}, \text{falsePosProb})$**
Third-party False Positive Risk

Positive response in public DNS

Delegated to querying organization in public DNS

Locally administered

None

1

0

0

1 - h

0

0

(next) (next)
Third-party False Positive Risk (cont’d)

• Remaining characteristics:
  • Name results in negative response from public DNS.
  • Name is not delegated to querying organization in public DNS.

• Consider each label in name, sorted hierarchically from top: \( L_n = [l_1, l_2, \ldots, l_q] \)

\[
\begin{align*}
\text{risk} & \leftarrow 0; \\
\text{foreach } l_i \in L_n \text{ do} & \\
& \quad \text{if } YX(l_i) \text{ then} \\
& \quad \quad \text{if } l_i \in LA \text{ then} \\
& \quad \quad \quad \text{risk} \leftarrow \text{risk} + (1 - h) \times \frac{0.5}{i} \\
& \quad \quad \text{else} \\
& \quad \quad \quad \text{risk} \leftarrow \text{risk} + \frac{0.5}{i} \\
& \quad \text{end} \\
& \text{end}
\end{align*}
\]
Third-party Leakage Risk

Delegated to querying organization in public DNS

None

Locally administered

1 - h
Third-party Collision Risk

\[ Q_{(c,S)} (n) = [n_1, n_2, \ldots, n_m] \]

- Third-party False Positive Risk
- Third-party Leakage Risk
- Importance factor \((I(n))\)

Aggregated like probabilities of independent events.

\[ I(n) \times \left( 1 - \left( \prod_{n_i \in [n_1, n_2, \ldots, n_m]} 1 - \text{thirdPartyFalsePosRisk}(n_i) \times (1 - \text{thirdPartyLeakageRisk}(n_m)) \right) \right) \]
Case Studies

• Case 1 – simple
  • Configuration
    • Search list: empty
    • Locally administered namespace: none
  • Query name: “foo.example”
    • DNS query names: [foo.example]
    • Result expected: positive or negative
  • Collision probability: 0
  • Third-party collision risk: 0
Case Studies

• Case 2 – simple search list
  • Configuration
    • Search list: [foo.example]
    • Locally administered namespace: none
  • Query name: “www”
    • DNS query names: [www.foo.example]
    • Result expected: positive
  • Collision probability: 0
  • Third-party collision risk: 0
Case Studies

- Case 3
  - Configuration
    - Search list: [foo.example]
    - Locally administered namespace: foo.example
      - foo.example delegated to third party
  - Query name: “www”
    - DNS query names: [www.foo.example]
    - Result expected: positive
  - Collision probability: 1 - h
  - Third-party collision risk: 1 - h
Case Studies

• Case 4
  • Configuration
    • Search list: [foo.example]
    • Locally administered namespace: foo.example
      • foo.example delegated to third party
  • Query name: “www”
    • DNS query names (depends on OS):
      • [www.foo.example]; or
      • [www.foo.example, www]
    • Result expected: negative
  • Collision probability: $1 - h$
  • Third-party collision risk: $1 - h$
Case Studies

• Case 5
  • Configuration
    • Search list: [foo.example]
    • Locally administered namespace: foo.example
      • foo.example delegated to third party
  • Query name: “www”
    • DNS query names (depends on OS):
      • [www.foo.example]; or
      • [www.foo.example, www]
  • Result expected: negative
    • “www” **delegated** in public DNS (but returns negative response)
  • Collision probability: 1
    • Third-party collision risk: \(1 - (1 - (1 - h))(1 - 0.5)(1 - h) = 1 - 0.5h^2\)
Case Studies

• Case 6
  • Configuration
    • Search list: [foo.example]
    • Locally administered namespace: foo.example
      • foo.example delegated to third party
  • Query name: “www”
    • DNS query names (depends on OS):
      • [www.foo.example]; or
      • [www.foo.example, www]
    • Result expected: negative
      • “www” delegated in public DNS and has positive response
  • Collision probability: 1
  • Third-party collision risk: 1
Model Application

- **Required:**
  - Suffix search list
  - Locally administered zones
- Derived from measurement/monitoring framework:
  - Queried names/results
  - Resolver configurations
  - Existence of names in public/private DNS

- **Variable:**
  - Locality
  - Existence of names in public DNS (e.g., in anticipation of future delegations)

- **Computed:**
  - Risk/potential
Summary

- Accurately quantifying name collision risk involves accurate modeling of resolver behavior “inside out”.
- Modeling provides foundation for metrics.
- Modeling leads to application.