A Method for Mitigating Namespace Collisions

Abstract

   This document outlines a possible, but not recommended, method to
   mitigate the effect of collisions in the DNS namespace by providing a
   means for end users to disambiguate the conflict.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7304.

Table of Contents

1. Introduction/Background

   Collisions in the DNS occur in multiple ways. One common case is
   that an organization has used a subdomain (foo) of its primary domain
   (example.com) for corporate infrastructure, and then the string 'foo'
   is delegated as a Top-Level Domain (TLD). When an employee of the
   organization enters 'www.foo', is the goal to reach a machine in the
   internal namespace (www.foo.example.com) or the hostname 'www' in the
   'foo' TLD?

   This document describes a means of disambiguating this and similar
   cases.

   Implementation of this method is not recommended; it is documented
   here to explain some of the pitfalls with such an approach.

2. Mitigation

   The mitigation described in this document involves presenting
   multiple options to the user and allowing them to indicate which of
   the names is the one they are trying to reach.

   The mitigation would look up the name in multiple namespaces. If a
   conflict is detected, it would then provide a means for the user to
   indicate which one of the colliding names they wish to connect to,
   and return the disambiguated answer to the application. An
   additional feature of mitigation could be to cache the user's choice
   and/or provide a means to set priorities.

   This could be accomplished in a number of ways, including:

   o Intercepting the resolution requests from the application in a
      "shim" type library

   o Replacing the resolver library entirely

   o Integrating this type of mitigation into applications (some web
      browsers already do something similar to this)

o Proxying the request to a server that provides an interstitial
   page that allows the user to indicate the intended name (for
   applications such as HTTP)

There are a number of issues with this solution, including but not
limited to:

o There may not be a human available to disambiguate the answer
   (unattended machines, mail servers, etc.).

o The human/user may have no idea which is the correct choice,
   especially in the case of a phishing attack or other malicious
   intent.

o The additional latency introduced may cause the originating
   application to time out.

o The experience would be time consuming for users as they must
   select each site and subsite intended (e.g., www.intranet,
   images.intranet, etc.).

o Caching the responses could lead to problems when the user changes
   location (internal sites would fail when offsite or otherwise lead
   to incorrect sites being loaded).

For these and other reasons, implementation of this technique is not
recommended.

3. Implementation/Disclaimers

   This document does not reference an implementation. Due to the
   numerous issues described above, we do not recommend that this
   solution be implemented. This is a very slight mitigation, and we do
   not recommend that it be viewed as a solution to the namespace
   collision problem.

4. Security Considerations

   While this method may make some users more aware of which version of
   a name they are going to use (and so careful users may avoid some
   phishing attacks), the security risks described above outweigh this
   potential benefit.

   There are numerous security implications in this approach, including
   leaking internal names (e.g., secret-project.corp.example.com), users
   being tricked into selecting the incorrect choice when trying to
   disambiguate answers, etc.

For these reasons, it is not recommended that this solution be
deployed.

5. Acknowledgements

The author wishes to thank the following individuals: Fred Baker, Bob
Braden, Carsten Bormann, Nevil Brownlee, Eric Burger, Brian
Carpenter, Benoit Claise, Keith Drage, Martin J. Duerst, David
Harrington, Paul Hoffamn, John Levine, and Ted Lemon.

Author's Address

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

EMail: warren@kumari.net