Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 5, 2012

A. Sullivan
Dyn, Inc.

May 4, 2012

Asserting Administrative Boundaries of Origin Using DNS Zones
draft-sullivan-domain-origin-assert-00

Abstract

Some clients on the Internet make inferences about the administrative relationships among servers on the Internet based on the domain names of those servers. Examples include decisions about acceptance of cookies and about cross-document information sharing in ECMAScript DOM. Perhaps unfortunately, real administrative boundaries in the DNS are not possible to detect, and therefore these inferences can go wrong in several ways. Mitigation strategies deployed so far will not scale. The solution to this is to provide a way to make an explicit assertion about the relationships between different domain names.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 5, 2012.

Table of Contents

1. Introduction

Many network resources are accessed primarily by name. DNS names make up a fundamental part of the name by which people or other systems access those network resources. As a result, DNS names have become fundamental elements in building security policies and user agent behaviour. Some such policies attempt to determine the scope for data sharing of things like HTTP state management cookies [RFC6265] and cross-document information sharing in ECMAScript DOM. The idea is to foil the attempts of attackers in (for example) attackersite.co.tld from setting cookies for everyone in co.tld.

Another use of the policy is a user interface convention that attempts to display the "real" domain name differently from other parts of the fully-qualified domain name, in an effort to decrease the success of phishing attacks. In this strategy, for instance, a domain name like "www.bank.example.com.attackersite.tld" is formatted to highlight that the name is inside "attackersite.tld", thereby hopefully reducing the user's impression that the user is visiting "www.bank.example.com".

Issuers of X.509 certificates make judgements about administrative boundaries around domains when issuing the certificates. For some discussion of the relationship between DNS names and X.509 certificates, see [RFC6125].

One way to build a reasonable policy is to treat each different domain name distinctly. Under this approach, foo.example.org, bar.example.org, and baz.example.org are all just different domains. Such an approach can be awkward, however, when (as is often the case) the real administrative boundary is a shared one (in this example, example.org). Therefore, clients have attempted to make more sophisticated policies.

Historically, some policies were based on the DNS tree. Early policies (for instance, in the earliest HTTP cookie specifications) just made a distinction between top-level domains and everything else; but this was too naive, and later attempts were based on inferences from the DNS names themselves. That did not work well, because there is no way in the DNS to discover the boundaries of administrative control around domain names.

Some have attempted to use the boundary of zone cuts (i.e. the location of the zone's apex and SOA record; see [RFC1034] and [RFC1035]). Unfortunately, that boundary is neither necessary nor sufficient for these purposes: it is possible for a large site to have many, administratively distinct subdomain-named sites without inserting an SOA record, and it is also possible that an

administrative entity (like a company) might divide its domain up into different zones for administrative reasons unrelated to the purposes of sites named in that domain. It was also, prior to the advent of DNSSEC, difficult to find zone cuts.

What appears to be needed is a mechanism to determine administrative boundaries in the DNS. That is, given two domain names, one needs to be able to answer whether the first name lies within the same administrative realm as the second. [[anchor2: I've used "administrative realm" here in an attempt to come up with yet another suitable term. "Administrative domain" is one other suggestion, though I fear a confusion between "administrative domain" and "domain" simpliciter, especially given that DNS operators are sometimes called domain administrators (so the domain is their administrative domain, of course, which is just confusing). Other thoughts on these terms welcome. --ajs@anvilwalrusden.com]]

A particularly important distinction for security purposes is the one between names that are mostly used to contain other domains, as compared to those that are mostly used to operate services. The former are often "delegation-centric" domains, delegating parts of their name space to others, and are frequently called "public suffix" domains. The term "public suffix" comes from a site, publicsuffix.org, which publishes a list of domains (henceforth, the "public suffix list") that are used to contain other domains. Not all, but most, delegation-centric domains are public suffix domains; and not all public suffix domains need to do DNS delegation, although most of them do. The reason for the public suffix list is to make the distinction between names that must never be treated as being in the same adminsitrative boundary, and those that may be so treated.

Unfortunately, the public suffix list has several inherent limitations. To begin with, it is a list that is separately maintained from the list of DNS delegations. As a result, the data in the public suffix list can diverge from the actual use of the DNS. Second, because its semantics are not the same as those of the DNS, it does not capture unusual features of the DNS, such as the empty non-terminal name. Third, as the size of the root zone grows, keeping the list both accurate and synchronized with the expanding services will become difficult and unreliable. Perhaps most importantly, it puts the power of assertion about the operational policies of a domain outside the control of the operators of that domain, and in the control of a third party possibly unrelated to those operators.

There have been suggestions for improvements of the public suffix list, most notably in [I-D.pettersen-subtld-structure]. It is unclear the extent to which those improvements would help, because

they represent improvements on the fundamental mechansism of keeping metadata about the DNS tree apart from the DNS tree itself.

2. Background and terminology

The reader is assumed to be familiar with the DNS ([RFC1034] [RFC1035]) and DNSSEC ([RFC4033] [RFC4034] [RFC4035] [RFC5155]).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Overview of mechanism

3.1. Records in the DNS

The basic mechanism uses resource records in the DNS to provide names through which the administrative boundary extends. The resource record is called BOUND (for administrative BOUNDary), RRTYPE TBD. [[anchor6: This could perhaps be a NAPTR or some such record with underscore conventions on the name, except that then I don't see how to make it extend well to underscore names themselves. Ideas? The disadvatage of a new RRTYPE is the reported difficulty of provisioning new RRs. --ajs@anvilwalrusden.com]]

Each BOUND resource record contains in its RDATA either one fully-qualified domain name, or a domain name containing the wildcard character "*" in the leftmost label, or the special string "_"; for more on the latter two, see Section 3.2.

There may be more than one BOUND resource record per name in the response. Each domain name in the RDATA is treated as a part of a common administrative realm as the owner name in the original QNAME.

There are three possible responses to a query for the BOUND RRTYPE at an owner name that are relevant to determining the administrative realm. The first is Name Error (RCODE=3, also known as NXDOMAIN). In this case, the name itself does not exist, and no further processing is needed.

The second is a No Data response [RFC2308] of any type. The No Data response means that the DNS named by the QNAME does not recognize any other name as part of a common administrative realm.

The final is a response with one or more BOUND resource records in the Answer section. Each BOUND resource record asserts that the name

identified in its RDATA shares the administrative realm of the owner name. The RDATA either contains a multi-label domain name relative to the root zone (see section 3.1 of [RFC1034]) or a string with some special characters in it (see Section 3.2.

Any other response is no different from any other sort of response from the DNS, and is not in itself meaningful for determining the administrative realm of a name (though it might be meaningful for finding the BOUND record).

3.2. Special target labels

3.2.1. Underscore target

A BOUND resource record with the single label "_" (called the "underscore target") is a positive assertion that no other domain name falls inside the administrative realm of the owner name. The record has a special use: it may be used to bootstrap operation. A client that has encountered the underscore target may remember the existence of the underscore target even after the expiry of the TTL on the resource record, until such time as a new query for the owner name may be made successfully. This rule permits implementations to cache positive statements of administrative isolation during disconnected periods, thereby starting a subsequent session with the values of prior affirmed policy. Apart from this bootstrapping use, and the ability of such an RR to have a TTL independent of the negative TTL value for the zone, this mechanism is semantically equivalent to a No Data answer.

It would be absurd for the underscore target to exist with any other BOUND resource record at that owner name. An authoritative name server MAY refuse to serve a zone containing such an inconsistency, MAY refuse to load a zone containing such an inconsistency, or MAY suppress every BOUND RR at an owner name except that containing the underscore target. The name server side of a recursive resolver MAY discard every BOUND RR at an owner name except that containing the underscore target. Conforming servers MUST NOT serve the underscore target and any other BOUND RR at the same owner name. Clients receiving a BOUND RRset that includes the underscore target MUST accept that RR, and discard any other RR in the RRset.

3.2.2. Wildcards in targets

The special character "*" is used to match any label, according to the wildcard label rules in section 4.3.3 of [RFC1034].

3.3. Wire format of the BOUND Resource Record

[[anchor8: To be provided if we decide that the BOUND RR is the right thing to do. --ajs@anvilwalrusden.com]]

4. An example case

For the purposes of discussion, it will be useful to imagine a portion of the DNS, using the domain example.tld. A diagram of the tree of this portion is in Figure 1. In the example, the domain example.tld includes several other names: www.example.tld, account.example.tld, cust1.example.tld, cust2.example.tld, test.example.tld, cust1.test.example.tld, and cust2.test.example.tld.

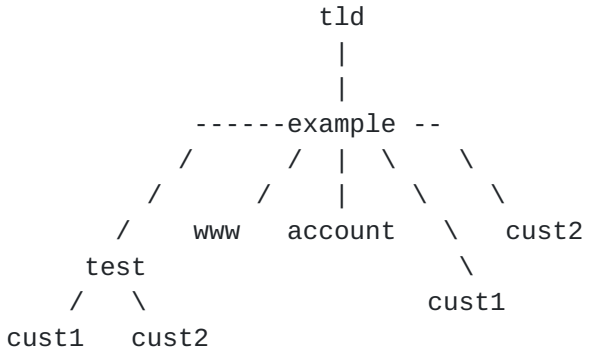

Figure 1

In the example, the domain tld delegates the domain example.tld. There are other possible cut points in the example, and depending on whether the cuts exist there may be implications for the use of the examples. See Section 4.1, below.

The (admittedly artificial) example permits us to distinguish a number of different roles. To begin with, there are three parties involved in the operation of services:

- OperatorV, the operator of example.tld;
- Operator1, the operator of cust1.example.tld;
- Operator2, the operator of cust2.example.tld.

Since there are three parties, there are likely three admininstrative boundaries as well; but the example contains some others. For instance, the names www.example.tld and example.tld are undoubtedly in the same administrative realm. By way of contrast, account.example.tld might be treated as completely separate, because OperatorV might wish to ensure that the accounts sytem is never permitted to share anything with any other name. By the same token, the names underneath test.example.tld are actually the test-instance

sites for customers. So cust1.test.example.tld might be in the same administrative realm as cust1.example.tld, but test.example.tld is certainly not in the same administrative realm as www.example.tld.

Finally, supposing that Operator1 and Operator2 merge their operations, it seems that it would be useful for cust1.example.tld and cust2.example.tld to lie in the same administrative realm, without including everything else in example.tld.

4.1. Examples of using the BOUND record for determining boundaries

This section provides some examples of different configurations of the example tree in Section 4, above. The examples are not exhaustive, but may provide an indication of what might be done with the mechanism.

4.1.1. One delegation, eight administrative realms, no underscore target

In this scenario, the example portion of the DNS name space contains all and only the following BOUND records:

```
example.tld 86400 IN BOUND www.example.tld
www.example.tld 86400 IN BOUND example.tld
```

Tld is the top-level domain, and has delegated example.tld. The operator of example.tld makes no delegations. There are four operators involved: the operator of tld, the operator of example.tld, the operator of the services at cust1.example.tld and cust1.test.example.tld, and the operator of the services at cust2.example.tld and cust2.test.example.tld.

In this arrangement, example.tld and www.example.tld positively claim to be within the same administrative realm. Every other name stands alone. A query for a BOUND record at any of those other names will result in a No Data response, which means that none of them include any other name in the same administrative realm. As a result, there are eight separate administrative realms in this case: tld, {example.tld and www.example.tld}, test.example.tld, cust1.test.example.tld, cust2.test.example.tld, account.example.tld, cust1.example.tld, and cust2.example.tld.

4.1.2. One delegation, eight administrative realms, underscore targets

This example mostly works the same way as the one in Section Section 4.1.1, but there is a slight difference. In this case, both tld and test.example.tld publish underscore targets in their BOUND records:

tld 86400 IN BOUND _
test.example.tld 86400 IN BOUND _

The practical effect of this is largely the same, except that these expressions of policy last 86,400 seconds instead of the length of time on the negative TTL in the relevant SOA for the zone. Many zones have short negative TTLs because of expectations that newly-added records will show up quickly. This mechanism permits such names to express their administrative isolation for predictable periods of time. Moreover, because clients are permitted to retain these records during periods when DNS service is not available, a client could go offline for several weeks, and return to service with the presumption that test.example.tld is still not in any administrative realm with any other name.

4.1.3. Two delegations, seven or eight administrative realms, underscore targets

In this scenario, example.tld delegates the name test.example.tld. In this case, there is an SOA record for test.example.tld. The BOUND record at test.example.tld is maintained, however.

At the same time, the Operator1 determines that it is safe to treat the test instance and production instance as being in the same adminsitrative realm. To begin with, Operator1 asks OperatorV to add the following record to the test.example.tld zone:

cust1.test.example.tld 86400 IN BOUND cust1.example.tld

This arrangement is not complete yet. Until a record is also added at cust1.example.tld, Operator1's intention is only half fulfilled. The service at cust1.test.example.tld treats cust1.example.tld as part of a common administrative realm, but the converse is not the case. [[anchor9: I can't decide whether there's anything useful in this configuration. Thoughts? --ajs@anvilwalrusden.com]]

To complete the process, Operator1 asks OperatorV to add the following record to the example.tld zone:

cust1.example.tld 86400 IN BOUND cust1.test.example.tld

Once this is complete, both names treat the other as part of the same administrative realm. In the end, the example segment of the DNS expresses the following administrative realms: tld, {example.tld, www.example.tld}, test.example.tld, {cust1.test.example.tld, cust1.example.tld}, cust2.example.tld, account.example.tld, cust2.example.tld.

5. Handling truncation

It is possible to put enough BOUND records into a zone such that the resulting response will exceed DNS or UDP protocol limits. In such cases, a UDP DNS response will arrive with the TC (truncation) bit set. Am BOUND response with the TC bit must be queried again in order to retrieve a complete response, in order to ensure that there is no missing underscore target (see Section 3.2.1).

6. Limitations of the approach

There are three significant problems with this proposal, all of which are related to using DNS to deliver the data.

The first is that new DNS RRTYPEs are difficult to deploy. While adding a new RRTYPE is straightforward, many provisioning systems do not have the necessary support and some firewalls and other edge systems continue to filter RRTYPEs they do not know.

The second is that it is difficult for an application to obtain data from the DNS. The TTL on an RRset, in particular, is usually not available to an application, even if the application uses the facilities of the operating system to deliver other parts of an unknown RRTYPE.

Finally, in many environments the system hosting the application has only proxied access to the Internet, and cannot query the DNS directly. It is not clear how such clients could ever possibly retrieve the BOUND record for a name.

7. Security Considerations

This mechanism enables publication of assertions about administrative relationships of different DNS-named systems on the Internet. If such assertions are accepted without checking that both sides agree to the assertion, it would be possible for one site to become an illegitimate source for data to be consumed in some other site.

Undertaking any of the inferences suggested in this draft without the use of the DNS Security Extensions exposes the user to the possibility of forged DNS responses.

8. IANA Considerations

IANA will be requested to register the BOUND RRTYPE if this proceeds.

9. Acknowledgements

The author thanks Dave Crocker, Jeff Hodges, Murray Kucherawy, Patrick McManus, Yngve N. Pettersen, and Peter St Andre for early discussion of this idea.

10. References

10.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

10.2. Informative References

[I-D.pettersen-subtld-structure] Pettersen, Y., "The Public Suffix Structure file format and its use for Cookie domain validation", draft-pettersen-subtld-structure-09 (work in progress), March 2012.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and

Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.

Author's Address

Andrew Sullivan
Dyn, Inc.
150 Dow St
Manchester, NH 03101
U.S.A.

Email: asullivan@dyn.com