Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2015

S. Cheshire
Apple Inc.

October 24, 2014

Special Use Top Level Domain "home"
draft-cheshire-homenet-dot-home-00

Abstract

This document specifies usage of the top-level domain "home", for names that are meaningful and resolvable within some scope smaller than the entire global Internet, but larger than the single link supported by Multicast DNS.

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 April 27, 2015.

Copyright Notice

1. Introduction

Globally unique domain names are available to individuals and organizations for a modest annual fee. However, there are situations where a globally unique domain name is not available, or has not yet been configured, and in these situations it is still desirable to be able to use DNS host names [RFC1034] [RFC1035], DNS-Based Service Discovery [RFC6763], and other DNS facilites.

In the absence of available globally unique domain names, Multicast DNS [RFC6762] makes it possible to use DNS facilities with names that are unique within the local link, using the "local" top-level domain.

This document specifies usage of a similar top-level domain, "home", for names that have scope larger than a single link, but smaller than the entire global Internet.

Author's Note [to be removed when document is published]: The purpose of this draft is not to propose some novel new usage for ".home" names. The purpose is to learn more about the current widespread use of ".home" names, and to document and formalize that usage.

Evidence [ICANN1][ICANN2] indicates that ".home" queries frequently leak out and reach the root name servers. We speculate that this is because of widespread usage of ".home" names in home networks, for example to name a printer "printer.home." When a user takes their laptop to a public Wi-Fi hotspot, attempts by that laptop to contact that printer result in fruitless ".home" queries to the root name servers. It would be beneficial for operators of public Wi-Fi hotspots to recognize and answer such queries locally, thereby reducing unnecessary load on the root name servers, and this document would give those operators the authority to do that. Readers who are aware of other usages of ".home" names, that are not compatible with the rules proposed here, are encouraged to contact the authors with information to help revise and improve this draft.

It is expected that the rules for ".home" names outlined here will also be suitable to meet the needs of the IETF HOMENET Working Group, though that is not the primary goal of this document. The primary goal of this draft is to understand and document the current usage. If the needs of the IETF HOMENET Working Group are not met by this document codifying the current de facto usage, then the Working Group may choose to reserve a different Special Use Domain Name [RFC6761] which does meet their needs. With luck that may not be necessary, and a single document may turn out to be sufficient to serve both purposes. In any case, the HOMENET Working Group is likely to be a good community in which to find knowledge about how ".home" names are currently used.

2. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

3. Mechanism

Typical residential home gateways configure their local clients via DHCP. In addition to the client's IP address, this DHCP configuration information typically also includes other configuration parameters, like the IP address of the recursive (caching) DNS server the client is to use, which is usually the home gateway's own address (the home gateway is also a DNS cache/relay).

For a home network consisting of just a single link (or several physical links bridged together to appear as a single logical link to IP) Multicast DNS [RFC6762] is sufficient for client devices to look up the dot-local host names of peers on the same home network, and perform DNS-Based Service Discovery (DNS-SD) [RFC6763] of services offered on that home network.

For a home network consisting of multiple links that are interconnected using IP-layer routing instead of link-layer bridging, link-local Multicast DNS alone is insufficient because link-local Multicast DNS requests, by design, do not cross between links. (This was a deliberate design choice for Multicast DNS, since even on a single link multicast traffic is expensive -- especially on Wi-Fi links -- and multiplying the amount of multicast traffic by flooding it across multiple links would make that problem even worse.) In this environment, unicast DNS requests (as facilated by use of ".home" names instead of ".local" names) should be used for cross-link name resolution and service discovery.

For residential home networks, Zero Configuration [ZC] operation is desirable. A client device learns the appropriate DNS-SD queries to perform, without requiring any manual configuration from the user, by sending domain enumeration queries [RFC6763] to its configured DNS server (typically the home gateway).

For organizations and individuals with registered globally unique domain names under their control, the answers to the domain enumeration queries SHOULD reference appropriate globally unique domain names. For example, at IETF meetings, domain enumeration queries [RFC6763] currently return the domain "meeting.ietf.org.", which is globally unique and under the control of the IETF. This

domain enumeration answer is configured manually by the IETF meeting network administrators.

When a suitable globally unique domain name is available, manual configuration of that name in a residential home gateway (or similar enterprise equipment) is appropriate. The network infrastructure then communicates that information to clients, without any additional manual configuration required on those clients.

However, many residential customers do not have any registered globally unique domain name available. This may be because they don’t want to pay the annual fee, or because they are unaware of the process for obtaining one, or because they are simply uninterested in having their own globally unique domain. This category also includes customers who intend to obtain a globally unique domain, but have not yet done so. For these users, it would be valuable to be able to perform cross-link name resolution and service discovery using unicast DNS without requiring a globally unique domain name.

To facilitate zero configuration operation, residential home gateways should be sold preconfigured with the default unicast domain name "home". This default unicast domain name is not globally unique, since many different residential home gateways will be using the name "home" at the same time, but is sufficient for useful operation within a small collection of links. Such residential home gateways SHOULD offer a configuration option to allow the default (non-unique) unicast domain name to be replaced with a globally unique domain name for cases where the customer has a globally unique domain available and wishes to use it.

This use of the the top-level domain "home" for private local use is not new. Many home gateways have been using the name this way for many years, and it remains in widespread use, as evidenced by the large volume of invalid queries for "home" reaching the root name servers [ICANN1][ICANN2]. The current root server traffic load is due to things like home gateways configuring clients with "home" as a search domain, and then leaking the resulting dot-home queries upstream. In large part what the document proposes is, "stop leaking dot-home queries upstream." This document codifies the existing practice, and provides formal grounds basis for ISPs to legitimately block such queries in order to reduce unnecessary load on the root name servers.

4. Security Considerations

Users should be aware that names in the "home" domain have only local significance. The name "My-Printer.home" in one location may not reference the same device as "My-Printer.home" in a different location.

5. IANA Considerations

[Once published, this should say] IANA has recorded the top-level domain "home" in the Special-Use Domain Names registry [SUDN].

5.1. Domain Name Reservation Considerations

The top-level domain "home", and any names falling within that domain (e.g. "My-Computer.home.", "My-Printer.home.", "_ipp._tcp.home."), are special [RFC6761] in the following ways:

1. Users may use these names as they would other DNS names, entering them anywhere that they would otherwise enter a conventional DNS name, or a dotted decimal IPv4 address, or a literal IPv6 address.

   Since there is no global authority responsible for assigning dot-home names, devices on different parts of the Internet could be using the same name. Users SHOULD be aware that using a name like "www.home" may not actually connect them to the web site they expected, and could easily connect them to a different web page, or even a fake or spoof of their intended web site, designed to trick them into revealing confidential information. As always with networking, end-to-end cryptographic security can be a useful tool. For example, when connecting with ssh, the ssh host key verification process will inform the user if it detects that the identity of the entity they are communicating with has changed since the last time they connected to that name.

2. Application software may use these names the same way it uses traditional globally unique unicast DNS names, and does not need to recognize these names and treat them specially in order to work correctly. This document specifies the use of the top-level domain "home" in on-the-wire messages. Ideally this would be purely a protocol-level identifier, not seen by end users. However, in some applications domain names are seen by end users, and in those cases, the protocol-level identifier "home" becomes visible, even for users for whom English is not their preferred language. For this reason, applications MAY choose to use additional UI cues (icon, text color, font, highlighting, etc.)

to communicate to the user that this is a special name with special properties. Due to the relative ease of spoofing dot-home names, end-to-end cryptographic security remains important when communicating across a local network, just as it is when communicating across the global Internet.

3. Name resolution APIs and libraries SHOULD NOT recognize these names as special and SHOULD NOT treat them differently. Name resolution APIs SHOULD send queries for these names to their configured recursive/caching DNS server(s).

4. Recursive/caching DNS servers SHOULD recognize these names as special and SHOULD NOT, by default, attempt to look up NS records for them, or otherwise query authoritative DNS servers in an attempt to resolve these names. Instead, recursive/caching DNS servers SHOULD, by default, act as authoritative and generate immediate responses for all such queries. This is to avoid unnecessary load on the root name servers and other name servers.

   The type of response generated depends on the role of the recursive/caching DNS server: (i) Traditional recursive DNS servers (such as those run by ISPs providing service to their customers) SHOULD, by default, generate immediate negative responses for all such queries. (ii) Recursive/caching DNS servers incorporated into residential home gateways of the kind described by this document should act as authoritative for these names and return positive or negative responses as appropriate.

   Recursive/caching DNS servers MAY offer a configuration option to enable upstream resolving of these names, for use in networks where these names are known to be handled by an authoritative DNS server in said private network. This option SHOULD be disabled by default, and SHOULD be enabled only when appropriate, to avoid queries leaking out of the private network and placing unnecessary load on the root name servers.

5. Traditional authoritative DNS servers SHOULD recognize these names as special and SHOULD, by default, generate immediate negative responses for all such queries, unless explicitly configured otherwise by the administrator. As described above, DNS servers incorporated into residential home gateways of the kind described by this document should act as authoritative for these names and return positive or negative responses as appropriate, unless explicitly configured otherwise by the administrator.

6. DNS server operators SHOULD, if they are using these names, configure their authoritative DNS servers to act as authoritative

for these names. In the case of zero-configuration residential home gateways of the kind described by this document, this configuration is implicit in the design of the product, rather than a result of conscious administration by the customer.

7. DNS Registries/Registrars MUST NOT grant requests to register these names in the normal way to any person or entity. These names are reserved for use in private networks and fall outside the set of names available for allocation by registries/registrars. Attempting to allocate a these name as if it were a normal DNS domain name will probably not work as desired, for reasons 4, 5, and 6 above.

6. Acknowledgments

Thanks to Francisco Arias of ICANN for his review and comments on this draft.

7. References

7.1. 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.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, February 2013.

7.2. Informative References

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[ICANN1] "New gTLD Collision Risk Mitigation", <https://www.icann.org/en/about/staff/security/ssr/new-gtld-collision-mitigation-05aug13-en.pdf>.

[ICANN2] "New gTLD Collision Occurrence Management", <https://www.icann.org/en/system/files/files/resolutions-new-gtld-annex-1-07oct13-en.pdf>.

[SUDN] "Special-Use Domain Names Registry", <http://www.iana.org/assignments/special-use-domain-names/>.

[ZC] Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Author's Address

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 14, 2015

S. Cheshire
Apple Inc.

November 10, 2014

Hybrid Unicast/Multicast DNS-Based Service Discovery
draft-ietf-dnssd-hybrid-00

Abstract

Performing DNS-Based Service Discovery using purely link-local Multicast DNS enables discovery of services that are on the local link, but not (without some kind of proxy or similar special support) of services that are outside the local link. Using a very large local link with thousands of hosts improves service discovery, but at the cost of large amounts of multicast traffic.

Performing DNS-Based Service Discovery using purely Unicast DNS is more efficient, but requires configuration of DNS Update keys on the devices offering the services, which can be onerous for simple devices like printers and network cameras.

Hence a compromise is needed, that provides easy service discovery without requiring either large amounts of multicast traffic or onerous configuration.

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 May 14, 2015.

Table of Contents

1. Introduction 3
2. Conventions and Terminology Used in this Document 4
3. Hybrid Proxy Operation 5
3.1. Domain Enumeration 6
3.2. Delegated Subdomain for LDH Host Names 7
3.3. Delegated Subdomain for Reverse Mapping 9
3.4. Data Translation 10
3.4.1. DNS TTL limiting 10
3.4.2. Suppressing Unusable Records 10
3.4.3. Application-Specific Data Translation 11
3.5. Answer Aggregation 12
3.5.1. Discovery of LLQ Service 14
4. Implementation Status 15
4.1. Already Implemented and Deployed 15
4.2. Partially Implemented 15
4.3. Not Yet Implemented 16
5. IPv6 Considerations 16
6. Security Considerations 17
6.1. Authenticity 17
6.2. Privacy 17
6.3. Denial of Service 17
7. Intelectual Property Rights 18
8. IANA Considerations 18
9. Acknowledgments 18
10. References 18
10.1. Normative References 18
10.2. Informative References 19
Author's Address 19

1. Introduction

Multicast DNS [RFC6762] and its companion technology DNS-based Service Discovery [RFC6763] were created to provide IP networking with the ease-of-use and autoconfiguration for which AppleTalk was well known [RFC6760] [ZC].

For a small network consisting of just a single link (or several physical links bridged together to appear as a single logical link to IP) Multicast DNS [RFC6762] is sufficient for client devices to look up the dot-local host names of peers on the same home network, and perform DNS-Based Service Discovery (DNS-SD) [RFC6763] of services offered on that home network.

For a larger network consisting of multiple links that are interconnected using IP-layer routing instead of link-layer bridging, link-local Multicast DNS alone is insufficient because link-local Multicast DNS packets, by design, do not cross between links. (This was a deliberate design choice for Multicast DNS, since even on a single link multicast traffic is expensive -- especially on Wi-Fi links -- and multiplying the amount of multicast traffic by flooding it across multiple links would make that problem even worse.) In this environment, Unicast DNS would be preferable to Multicast DNS. (Unicast DNS can be used either with a traditionally assigned globally unique domain name, or with a private local unicast domain name such as ".home" [HOME].)

To use Unicast DNS, the names of hosts and services need to be made available in the Unicast DNS namespace. In the DNS-SD specification [RFC6763] Section 10 ("Populating the DNS with Information") discusses various possible ways that a service's PTR, SRV, TXT and address records can make their way into the Unicast DNS namespace, including manual zone file configuration [RFC1034] [RFC1035], DNS Update [RFC2136] [RFC3007] and proxies of various kinds.

This document specifies a type of proxy called a Hybrid Proxy that uses Multicast DNS [RFC6762] to discover Multicast DNS records on its local link, and makes corresponding DNS records visible in the Unicast DNS namespace.

2. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

The Hybrid Proxy builds on Multicast DNS, which works between hosts on the same link. A set of hosts is considered to be "on the same link" if:

- when any host A from that set sends a packet to any other host B in that set, using unicast, multicast, or broadcast, the entire link-layer packet payload arrives unmodified, and
- a broadcast sent over that link by any host from that set of hosts can be received by every other host in that set

The link-layer *header* may be modified, such as in Token Ring Source Routing [802.5], but not the link-layer *payload*. In particular, if any device forwarding a packet modifies any part of the IP header or IP payload then the packet is no longer considered to be on the same link. This means that the packet may pass through devices such as repeaters, bridges, hubs or switches and still be considered to be on the same link for the purpose of this document, but not through a device such as an IP router that decrements the IP TTL or otherwise modifies the IP header.

3. Hybrid Proxy Operation

In its simplest form, each physical link in an organization is assigned a unique Unicast DNS domain name, such as "Building 1.example.com" or "4th Floor.Building 1.example.com". Grouping multiple links under a single Unicast DNS domain name is to be specified in a future companion document, but for the purposes of this document, assume that each link has its own unique Unicast DNS domain name. In a graphical user interface these names are not displayed as strings with dots as shown above, but something more akin to a typical file browser graphical user interface (which is harder to illustrate in a text-only document) showing folders, subfolders and files in a file system.

Each named link in an organization has a Hybrid Proxy which serves it. This Hybrid Proxy function could be performed by a router on that link, or, with appropriate VLAN configuration, a single Hybrid Proxy could have a logical presence on, and serve as the Hybrid Proxy for, many links. In the parent domain, NS records are used to delegate ownership of each defined link name (e.g., "Building 1.example.com") to the Hybrid Proxy that serves the named link. In other words, the Hybrid Proxy is the authoritative name server for that subdomain.

When a DNS-SD client issues a Unicast DNS query to discover services in a particular Unicast DNS subdomain (e.g., "_printer._tcp.Building 1.example.com. PTR ?") the normal DNS delegation mechanism results in that query being forwarded until it reaches the delegated authoritative name server for that subdomain, namely the Hybrid Proxy on the link in question. Like a conventional Unicast DNS server, a Hybrid Proxy implements the usual Unicast DNS protocol [RFC1034] [RFC1035] over UDP and TCP. However, unlike a conventional Unicast DNS server that generates answers from the data in its manually-configured zone file, a Hybrid Proxy generates answers using Multicast DNS. A Hybrid Proxy does this by consulting its Multicast DNS cache and/or issuing Multicast DNS queries for the corresponding Multicast DNS name, type and class, (e.g., in this case, "_printer._tcp.local. PTR ?"). Then, from the received Multicast DNS data, the Hybrid Proxy synthesizes the appropriate Unicast DNS response.

Naturally, the existing Multicast DNS caching mechanism is used to avoid issuing unnecessary Multicast DNS queries on the wire. The Hybrid Proxy is acting as a client of the underlying Multicast DNS subsystem, and benefits from the same caching and efficiency measures as any other client using that subsystem.

3.1. Domain Enumeration

The administrator creates Domain Enumeration PTR records [RFC6763] to inform clients of available service discovery domains, e.g.,:

```
    b._dns-sd._udp.example.com.    PTR   Building 1.example.com.
                                   PTR   Building 2.example.com.
                                   PTR   Building 3.example.com.
                                   PTR   Building 4.example.com.

    db._dns-sd._udp.example.com.   PTR   Building 1.example.com.

    lb._dns-sd._udp.example.com.   PTR   Building 1.example.com.
```

The "b" ("browse") records tell the client device the list of browsing domains to display for the user to select from and the "db" ("default browse") record tells the client device which domain in that list should be selected by default. The "lb" ("legacy browse") record tells the client device which domain to automatically browse on behalf of applications that don't implement UI for multi-domain browsing (which is most of them, today). The "lb" domain is usually the same as the "db" domain.

DNS responses are limited to a maximum size of 65535 bytes. This limits the maximum number of domains that can be returned for a Domain Enumeration query, as follows:

A DNS response header is 12 bytes. That's typically followed by a single qname (up to 256 bytes) plus qtype (2 bytes) and qclass (2 bytes), leaving 65275 for the Answer Section.

An Answer Section Resource Record consists of:

- Owner name, encoded as a two-byte compression pointer
- Two-byte rrtype (type PTR)
- Two-byte rrclass (class IN)
- Four-byte ttl
- Two-byte rdlength
- rdata (domain name, up to 256 bytes)

This means that each Resource Record in the Answer Section can take up to 268 bytes total, which means that the Answer Section can contain, in the worst case, no more than 243 domains.

In a more typical scenario, where the domain names are not all maximum-sized names, and there is some similarity between names so that reasonable name compression is possible, each Answer Section Resource Record may average 140 bytes, which means that the Answer Section can contain up to 466 domains.

3.2. Delegated Subdomain for LDH Host Names

The rules for DNS-SD service instance names and domains are more permissive than the traditional rules for host names.

Users typically interact with DNS-SD by viewing a list of discovered service instance names on the display and selecting one of them by pointing, touching, or clicking. Similarly, in software that provides a multi-domain DNS-SD user interface, users view a list of offered domains on the display and select one of them by pointing, touching, or clicking. To use a service, users don't have to remember domain or instance names, or type them; users just have to be able to recognize what they see on the display and click on the thing they want.

In contrast, host names are often remembered and typed. Also, host names are often used in command-line interfaces where spaces can be inconvenient. For this reason, host names have traditionally been restricted to letters, digits and hyphens, with no spaces or other punctuation.

While we still want to allow rich text for DNS-SD service instance names and domains, it is advisable, for maximum compatibility with existing software, to restrict host names to the traditional letter-digit-hyphen rules. This means that while a service name "My Printer._ipp._tcp.Building 1.example.com" is acceptable and desirable (it is displayed in a graphical user interface as an instance called "My Printer" in the domain "Building 1" at "example.com"), a host name "My-Printer.Building 1.example.com" is not advisable (because of the space in "Building 1").

To accomodate this difference in allowable characters, a Hybrid Proxy MUST support having two subdomains delegated to it, one to be used for host names (names of 'A' and 'AAAA' address records), which is restricted to the traditional letter-digit-hyphen rules, and another to be used for other records (including the PTR, SRV and TXT records used by DNS-SD), which is allowed to be arbitrary Net-Unicode text [RFC5198].

For example, a Hybrid Proxy could have the two subdomains "Building 1.example.com" and "bldg1.example.com" delegated to it. The Hybrid Proxy would then translate these two Multicast DNS records:

```
My Printer._ipp._tcp.local. SRV 0 0 631 prnt.local.
prnt.local.                 A   10.0.1.2
```

into Unicast DNS records as follows:

```
My Printer._ipp._tcp.Building 1.example.com.
                            SRV 0 0 631 prnt.bldg1.example.com.
prnt.bldg1.example.com.     A   10.0.1.2
```

Note that the SRV record name is translated using the rich-text domain name ("Building 1.example.com") and the address record name is translated using the LDH domain ("bldg1.example.com").

3.3. Delegated Subdomain for Reverse Mapping

A Hybrid Proxy can facilitate easier management of reverse mapping domains, particularly for IPv6 addresses where manual management may be more onerous than it is for IPv4 addresses.

To achieve this, in the parent domain, NS records are used to delegate ownership of the appropriate reverse mapping domain to the Hybrid Proxy. In other words, the Hybrid Proxy becomes the authoritative name server for the reverse mapping domain.

For example, if a given link is using the IPv4 subnet 10.1/16, then the domain "1.10.in-addr.arpa" is delegated to the Hybrid Proxy for that link.

If a given link is using the IPv6 prefix 2001:0DB8/32, then the domain "8.b.d.0.1.0.0.2.ip6.arpa" is delegated to the Hybrid Proxy for that link.

When a reverse mapping query arrives at the Hybrid Proxy, it issues the identical query on its local link as a Multicast DNS query. (In the Apple "/usr/include/dns_sd.h" APIs, using ForceMulticast indicates that the DNSServiceQueryRecord() call should perform the query using Multicast DNS.) When the host owning that IPv4 or IPv6 address responds with a name of the form "something.local", the Hybrid Proxy rewrites that to use its configured LDH host name domain instead of "local" and returns the response to the caller.

For example, a Hybrid Proxy with the two subdomains "1.10.in-addr.arpa" and "bldg1.example.com" delegated to it would translate this Multicast DNS record:

```
3.2.1.10.in-addr.arpa. PTR prnt.local.
```

into this Unicast DNS response:

```
3.2.1.10.in-addr.arpa. PTR prnt.bldg1.example.com.
```

Subsequent queries for the prnt.bldg1.example.com address record, falling as it does within the bldg1.example.com domain, which is delegated to the Hybrid Proxy, will arrive at the Hybrid Proxy, where they are answered by issuing Multicast DNS queries and using the received Multicast DNS answers to synthesize Unicast DNS responses, as described above.

3.4. Data Translation

Generating the appropriate Multicast DNS queries involves, at the very least, translating from the configured DNS domain (e.g., "Building 1.example.com") on the Unicast DNS side to "local" on the Multicast DNS side.

Generating the appropriate Unicast DNS responses involves translating back from "local" to the configured DNS Unicast domain.

Other beneficial translation and filtering operations are described below.

3.4.1. DNS TTL limiting

For efficiency, Multicast DNS typically uses moderately high DNS TTL values. For example, the typical TTL on DNS-SD PTR records is 75 minutes. What makes these moderately high TTLs acceptable is the cache coherency mechanisms built in to the Multicast DNS protocol which protect against stale data persisting for too long. When a service shuts down gracefully, it sends goodbye packets to remove its PTR records immediately from neighbouring caches. If a service shuts down abruptly without sending goodbye packets, the Passive Observation Of Failures (POOF) mechanism described in Section 10.5 of the Multicast DNS specification [RFC6762] comes into play to purge the cache of stale data.

A Unicast DNS client on a remote link does not get to participate in these Multicast DNS cache coherency mechanisms on the local link. For Unicast DNS requests received without any LLQ option the DNS TTLs reported in the resulting Unicast DNS response SHOULD be capped to be no more than ten seconds. For received Unicast DNS requests that contain an LLQ option, the Multicast DNS record’s TTL SHOULD be returned unmodified, because the LLQ notification channel exists to inform the remote client as records come and go. For further details about the LLQ option, see Section 3.5.

3.4.2. Suppressing Unusable Records

A Hybrid Proxy SHOULD suppress Unicast DNS answers for records that are not useful outside the local link. For example, DNS A and AAAA records for IPv4 link-local addresses [RFC3927] and IPv6 link-local addresses [RFC4862] should be suppressed. Similarly, for sites that have multiple private address realms [RFC1918], private addresses from one private address realm should not be communicated to clients in a different private address realm.

By the same logic, DNS SRV records that reference target host names

that have no addresses usable by the requester should be suppressed, and likewise, DNS PTR records that point to unusable SRV records should be similarly be suppressed.

3.4.3. Application-Specific Data Translation

There may be cases where Application-Specific Data Translation is appropriate.

For example, AirPrint printers tend to advertise fairly verbose information about their capabilities in their DNS-SD TXT record. This information is a legacy from LPR printing, because LPR does not have in-band capability negotiation, so all of this information is conveyed using the DNS-SD TXT record instead. IPP printing does have in-band capability negotiation, but for convenience printers tend to include the same capability information in their IPP DNS-SD TXT records as well. For local mDNS use this extra TXT record information is inefficient, but not fatal. However, when a Hybrid Proxy aggregates data from multiple printers on a link, and sends it via unicast (via UDP or TCP) this amount of unnecessary TXT record information can result in large responses. Therefore, a Hybrid Proxy that is aware of the specifics of an application-layer protocol such as Apple's AirPrint (which uses IPP) can elide unnecessary key/value pairs from the DNS-SD TXT record for better network efficiency.

Note that this kind of Application-Specific Data Translation is expected to be very rare. It is the exception, rather than the rule. This is an example of a common theme in computing. It is frequently the case that it is wise to start with a clean, layered design, with clear boundaries. Then, in certain special cases, those layer boundaries may be violated, where the performance and efficiency benefits outweigh the inelegance of the layer violation.

As in other similar situations, these layer violations optional. They are done only for efficiency reasons, and are not required for correct operation. A Hybrid Proxy can operate solely at the mDNS layer, without any knowledge of semantics at the DNS-SD layer or above.

3.5. Answer Aggregation

In a simple analysis, simply gathering multicast answers and forwarding them in a unicast response seems adequate, but it raises the question of how long the Hybrid Proxy should wait to be sure that it has received all the Multicast DNS answers it needs to form a complete Unicast DNS response. If it waits too little time, then it risks its Unicast DNS response being incomplete. If it waits too long, then it creates a poor user experience at the client end. In fact, there may no time which is both short enough to produce a good user experience and at the same time long enough to reliably produce complete results.

Similarly, the Hybrid Proxy -- the authoritative name server for the subdomain in question -- needs to decide what DNS TTL to report for these records. If the TTL is too long then the recursive (caching) name servers issuing queries on behalf of their clients risk caching stale data for too long. If the TTL is too short then the amount of network traffic will be more than necessary. In fact, there may no TTL which is both short enough to avoid undesirable stale data and at the same time long enough to be efficient on the network.

These dilemmas are solved by use of DNS Long-Lived Queries (DNS LLQ) [I-D.sekar-dns-llq]. The Hybrid Proxy responds immediately to the Unicast DNS query using the Multicast DNS records it already has in its cache (if any). This provides a good client user experience by providing a near-instantaneous response. Simultaneously, the Hybrid Proxy issues a Multicast DNS query on the local link to discover if there are any additional Multicast DNS records it did not already know about. Should additional Multicast DNS responses be received, these are then delivered to the client using DNS LLQ update messages. The timeliness of such LLQ updates is limited only by the timeliness of the device responding to the Multicast DNS query. If the Multicast DNS device responds quickly, then the LLQ update is delivered quickly. If the Multicast DNS device responds slowly, then the LLQ update is delivered slowly. The benefit of using LLQ is that the Hybrid Proxy can respond promptly because it doesn’t have to delay its unicast response to allow for the expected worst-case delay for receiving all the Multicast DNS responses. Even if a proxy were to try to provide reliability by assuming an excessively pessimistic worst-case time (thereby giving a very poor user experience) there would still be the risk of a slow Multicast DNS device taking even longer than that (e.g, a device that is not even powered on until ten seconds after the initial query is received) resulting in incomplete responses. Using LLQs solves this dilemma: even very late responses are not lost; they are delivered in subsequent LLQ update messages.

There are two factors that determine specifically how responses are generated:

The first factor is whether the query from the client included the LLQ option (typical with long-lived service browsing PTR queries) or not (typical with one-shot operations like SRV or address record queries). Note that queries containing the LLQ option are received directly from the client (see Section 3.5.1). Queries containing no LLQ option are generally received via the client's configured recursive (caching) name server.

The second factor is whether the Hybrid Proxy already has at least one record in its cache that positively answers the question.

- No LLQ option; no answer in cache:
  Do local mDNS query up to three times, return answers if received, otherwise return negative response if no answer after three tries.
  DNS TTLs in responses are capped to at most ten seconds.

- No LLQ option; at least one answer in cache:
  Send response right away to minimise delay.
  DNS TTLs in responses are capped to at most ten seconds.
  No local mDNS queries are performed.
  (Reasoning: Given RRSet TTL harmonisation, if the proxy has one Multicast DNS answer in its cache, it can reasonably assume that it has all of them.)

- Query contains LLQ option; no answer in cache:
  As above, do local mDNS query up to three times, and return answers if received.
  If no answer after three tries, return negative response.
  (Reasoning: We don't need to rush to send an empty answer.)
  In both cases the query remains active for as long as the client maintains the LLQ state, and if mDNS answers are received later, LLQ update messages are sent.
  DNS TTLs in responses are returned unmodified.

- Query contains LLQ option; at least one answer in cache:
  As above, send response right away to minimise delay.
  The query remains active for as long as the client maintains the LLQ state, and if additional mDNS answers are received later, LLQ update messages are sent.
  (Reasoning: We want UI that is displayed very rapidly, yet continues to remain accurate even as the network environment changes.)
  DNS TTLs in responses are returned unmodified.

Note that the "negative responses" referred to above are "no error no

answer" negative responses, not NXDOMAIN. This is because the Hybrid Proxy cannot know all the Multicast DNS domain names that may exist on a link at any given time, so any name with no answers may have child names that do exist, making it an "empty nonterminal" name.

3.5.1. Discovery of LLQ Service

To issue LLQ queries, clients need to communicate directly with the authoritative Hybrid Proxy. The procedure by which the client locates the authoritative Hybrid Proxy is described in the LLQ specification [I-D.sekar-dns-llq].

Briefly, the procedure is as follows: To discover the LLQ service for a given domain name, a client first performs DNS zone apex discovery, and then, having discovered <apex>, the client then issues a DNS query for the SRV record with the name _dns-llq._udp.<apex> to find the target host and port for the LLQ service for that zone. By default LLQ service runs on port 5352, but since SRV records are used, the LLQ service can be offered on any port.

A client performs DNS zone apex discovery using the procedure below:

1. The client issues a DNS query for the SOA record with the given domain name.

2. A conformant recursive (caching) name server will either send a positive response, or a negative response containing the SOA record of the zone apex in the Authority Section.

3. If the name server sends a negative response that does not contain the SOA record of the zone apex, the client trims the first label off the given domain name and returns to step 1 to try again.

By this method, the client iterates until it learns the name of the zone apex, or (in pathological failure cases) reaches the root and gives up.

Normal DNS caching is used to avoid repetitive queries on the wire.

4. Implementation Status

Some aspects of the mechanism specified in this document already exist in deployed software. Some aspects are new. This section outlines which aspects already exist and which are new.

4.1. Already Implemented and Deployed

Domain enumeration by the client (the "b._dns-sd._udp" queries) is already implemented and deployed.

Unicast queries to the indicated discovery domain is already implemented and deployed.

These are implemented and deployed in Mac OS X 10.4 and later (including all versions of Apple iOS, on all iPhone and iPads), in Bonjour for Windows, and in Android 4.1 "Jelly Bean" (API Level 16) and later.

Domain enumeration and unicast querying have been used for several years at IETF meetings to make Terminal Room printers discoverable from outside the Terminal room. When you Press Cmd-P on your Mac, or select AirPrint on your iPad or iPhone, and the Terminal room printers appear, that is because your client is doing unicast DNS queries to the IETF DNS servers.

4.2. Partially Implemented

The current APIs make multiple domains visible to client software, but most client UI today lumps all discovered services into a single flat list. This is largely a chicken-and-egg problem. Application writers were naturally reluctant to spend time writing domain-aware UI code when few customers today would benefit from it. If Hybrid Proxy deployment becomes common, then application writers will have a reason to provide better UI. Existing applications will work with the Hybrid Proxy, but will show all services in a single flat list. Applications with improved UI will group services by domain.

The Long-Lived Query mechanism [I-D.sekar-dns-llq] referred to in this specification exists and is deployed, but has not been standardized by the IETF. It is possible that the IETF may choose to standardize a different or better Long-Lived Query mechanism. In that case, the pragmatic deployment approach would be for vendors to produce Hybrid Proxies that implement both the deployed Long-Lived Query mechanism [I-D.sekar-dns-llq] (for today’s clients) and a new IETF Standard Long-Lived Query mechanism (as the future long-term direction).

The translating/filtering Hybrid Proxy specified in this document. Implementations are under development, and operational experience with these implementations has guided updates to this document.

4.3. Not Yet Implemented

A mechanism to 'stitch' together multiple ".local." zones so that they appear as one. Such a mechanism will be specified in a future companion document.

5. IPv6 Considerations

An IPv4-only host and an IPv6-only host behave as "ships that pass in the night". Even if they are on the same Ethernet, neither is aware of the other's traffic. For this reason, each physical link may have *two* unrelated ".local." zones, one for IPv4 and one for IPv6. Since for practical purposes, a group of IPv4-only hosts and a group of IPv6-only hosts on the same Ethernet act as if they were on two entirely separate Ethernet segments, it is unsurprising that their use of the ".local." zone should occur exactly as it would if they really were on two entirely separate Ethernet segments.

It will be desirable to have a mechanism to 'stitch' together these two unrelated ".local." zones so that they appear as one. Such mechanism will need to be able to differentiate between a dual-stack (v4/v6) host participating in both ".local." zones, and two different hosts, one IPv4-only and the other IPv6-only, which are both trying to use the same name(s). Such a mechanism will be specified in a future companion document.

6. Security Considerations

6.1. Authenticity

A service proves its presence on a link by its ability to answer link-local multicast queries on that link. If greater security is desired, then the Hybrid Proxy mechanism should not be used, and something with stronger security should be used instead, such as authenticated secure DNS Update [RFC2136] [RFC3007].

6.2. Privacy

The Domain Name System is, generally speaking, a global public database. Records that exist in the Domain Name System name hierarchy can be queried by name from, in principle, anywhere in the world. If services on a mobile device (like a laptop computer) are made visible via the Hybrid Proxy mechanism, then when those services become visibile in a domain such as "My House.example.com" that might indicate to (potentially hostile) observers that the mobile device is in my house. When those services disappear from "My House.example.com" that change could be used by observers to infer when the mobile device (and possibly its owner) may have left the house. The privacy of this information may be protected using techniques like firewalls and split-view DNS, as are customarily used today to protect the privacy of corporate DNS information.

6.3. Denial of Service

A remote attacker could use a rapid series of unique Unicast DNS queries to induce a Hybrid Proxy to generate a rapid series of corresponding Multicast DNS queries on one or more of its local links. Multicast traffic is expensive -- especially on Wi-Fi links -- which makes this attack particularly serious. To limit the damage that can be caused by such attacks, a Hybrid Proxy (or the underlying Multicast DNS subsystem which it utilizes) MUST implement Multicast DNS query rate limiting appropriate to the link technology in question. For Wi-Fi links the Multicast DNS subsystem SHOULD NOT issue more than 20 Multicast DNS query packets per second. On other link technologies like Gigabit Ethernet higher limits may be appropriate.

7. Intelectual Property Rights

Apple has submitted an IPR disclosure concerning the technique proposed in this document. Details are available on the IETF IPR disclosure page [IPR2119].

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

Thanks to Markus Stenberg for helping develop the policy regarding the four styles of unicast response according to what data is immediately available in the cache. Thanks to Andrew Yourtchenko for comments about privacy issues. [Partial list; more names to be added.]

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.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

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

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, May 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, December 2012.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, December 2012.

[I-D.sekar-dns-llq] Sekar, K., "DNS Long-Lived Queries", draft-sekar-dns-llq-01 (work in progress), August 2006.

10.2. Informative References

[HOME] Cheshire, S., "Special Use Top Level Domain 'home'", draft-cheshire-homenet-dot-home (work in progress), November 2014.

[IPR2119] "Apple Inc.'s Statement about IPR related to Hybrid Unicast/Multicast DNS-Based Service Discovery", <https://datatracker.ietf.org/ipr/2119/>.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol to Replace the AppleTalk Name Binding Protocol (NBP)", RFC 6760, December 2012.

[ZC] Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Author's Address

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

DNSSD
INTERNET-DRAFT
Intended Status: Informational
Expires: April 27, 2015

H. Rafiee

October 27, 2014

Multicast DNS (mDNS) Threat Model and Security Consideration
<draft-rafiee-dnssd-mdns-threatmodel-01.txt>

Abstract

This document describes threats associated with extending multicast DNS (mDNS) across layer 3.

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 April 27, 2015.

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 3
3. Threat Analysis 4
3.1. DoS attack on any node in the DNS-SD enabled network . . 4
3.1.1. Personal Area Network (PAN) 4
3.1.2. Temporary Public Hotspot 5
3.2. Node compromising 5
3.2.1. Home, Enterprise, Mesh networks 5
3.3. Spoofing Attacks & forge the Identity 5
3.3.1. Public Hotspot, Home, Enterprise, Mesh networks . . . 5
3.3.2. Enterprise network 5
3.4. Malicious update on unicast DNS 5
3.5. Cache Poisoning 6
3.6. Harming Privacy 6
3.7. Resource spoofing 6
3.8. Dual stack attacks 6
3.9. MAC address spoofing 6
3.10. Privacy Protection Mechanisms 6
3.10.1. The Use of Random Data 6
3.10.2. Data Encryption 7
3.11. Authorization of a Service Requester 7
3.11.1. The Use of an Access List 7
3.11.1.1. SAVI-DHCP 7
3.11.1.2. CGA-TSIG 7
3.11.1.3. DNS over DTLS 8
3.11.2. The Use of Shared Secret 8
3.12. Authorization of a Service Provider 8
3.12.1. SAVI-DHCP 8
3.12.2. Router advertisement 8
3.13. Other Security Considerations 8
3.14. Not Usable Security Mechanisms 9
3.14.1. DNSSEC 9
3.14.2. IPsec 9
4. Security Considerations 9
5. IANA Considerations 9
6. Acknowledgements 9
7. References 9
7.1. Normative 9
7.2. Informative 10
Authors' Addresses 11

1. Introduction

Multicast DNS (mDNS) was proposed in [RFC6762] to allow nodes in local links to use DNS-like names for their communication without the need for global DNS servers, infrastructure and administration processes for configuration. mDNS along with service discovery (DNS-SD) [RFC6763] provides nodes with the possibility to discover other services and the names of other nodes with zero configuration, i.e., connect a node into a local link and use resources such as a printer that are available in that network.

mDNS and service discovery (SD) use DNS- like query messages. The main assumption is that these services also use DNS security protocols such as DNSSEC. However, it cannot use DNSSEC for security because DNSSEC is not zero configuration service. This is why the current implementations use no security in local links and are vulnerable to several attacks.

The purpose of this document is to introduce threat models for service discovery and allow implementers to be aware of the possible attacks in order to mitigate them with possible solutions. Since there are already old lists of known DNS threats available in [RFC3833], here we only analyze the ones that are applicable to DNS-SD. We also introduce new possible threats that could result from extending DNS-SD scope.

2. Terminology

Node: any host and routers in the network

Attack: an action to exploit a node and allow the attacker to gain access to that node. It can be also an action to prevent a node from providing a service or using a service on the network

Attacker: a person who uses any node in the network to attack other nodes using known or unknown threats

Threat: Anything that has a potential to harm a node in the network

Local link vulnerability: Any flaws that are the result of the assumption that a malicious node could gain access to legitimate nodes inside a local link network

Wide Area Network (WAN) vulnerability: Any flaws that are the result of the assumption that a malicious node could gain access to legitimate nodes inside any local links in an enterprise network with multiple Local Area Networks (LANs) or Virtual LANs (VLANs).

Host name: Fully qualified DNS Name (FQDN) of a node in the network

Constrained device: a small device with limited resources (battery,

memory, etc.)

Service Providers: a node that offer a service to other nodes. One example of a service provider in DNS-SD is a printer.

Service Requester: a node in the network that requests a service by the use of DNS-SD protocols. One example of service requester is a computer that discovers a printer in the network and tries to use it.

3. Threat Analysis

DNS-SD cannot use DNSSEC approaches for security purposes. This is because, as mentioned earlier, DNSSEC is not a zero config protocol and it is not compatible with the plug and play nature of DNS-SD. This is why DNS-SD is vulnerable to several attacks. Most threats in this section are a result of spoofing, Denial of Service (DoS), or a combination of them. Here we explain them in different example scenarios. The definition of different use case scenarios are defined in [requirement].

There are several scenarios associated with the Large Traffic Production case.

First scenario: a malicious node in any of the subnets that the gateway connects can advertise different fake services or spoof the information of the real services and replay the messages. This causes large traffic either in the local link or in other links since the gateway was also supposed to replicate the traffic to other links.

Second scenario : a malicious node spoofs the legitimate service advertisements of different nodes in the network and changes the Time To Leave (TTL) value to zero. This will result in producing large traffic since the mDNS gateway needs to ask all of the service advertisers to re-advertise their service. This is an especially effective attack in a network of constrained devices because it causes more energy consumption.

3.1. DoS attack on any node in the DNS-SD enabled network

3.1.1. Personal Area Network (PAN)

When service provider and service requester are connected via a network cable or USB, then the only threat is virus or other malware that might infect any of these nodes. This might cause DoS.

Wireless PAN (WPAN) is where service provider and service requester are connected via Bluetooth or wireless. Since WPANs are short range and their coverage are usually limited, the attacker should be so close to any of those nodes to be able to perform any attacks. If this happens, the attacker might be able to forge the identity of the

service provider or perform DoS attack.

3.1.2. Temporary Public Hotspot

A malicious node can spoof the source IP address of a legitimate victim node and question several services in the link. This will result in a large traffic return to the victim node from both gateway and also service owner.

3.2. Node compromising

3.2.1. Home, Enterprise, Mesh networks

When ISP, home router/gateway and service provider (like a printer) support IPv6 address, then service providers usually automatically sets an IPv6 address. Since this address is global, this node is accessible over the internet. If the address of this service provider is known to the attacker, then it might be able to compromise this service provider and access to this network (because service providers usually supports weak security features).

3.3. Spoofing Attacks & forge the Identity

3.3.1. Public Hotspot, Home, Enterprise, Mesh networks

Scenario 1: A malicious node can spoof the source IP address of a legitimate victim node advertises fake services in the network. This might result in compromising the victim nodes or having malicious access to the victim nodes' resources.

Scenario2: A malicious node spoofs the content of Dynamic Host Configuration Protocol (DHCP) server messages and offers its own malicious information to the nodes in the network.

3.3.2. Enterprise network

A virus or any malware can compromise a legitimate node in this network. Then this node can forge the identity of service providers or perform DoS attack on this network.

3.4. Malicious update on unicast DNS

A malicious node can spoof the content of DNS update message and add malicious records to unicast DNS. This attack is applicable on enterprise networks.

3.5. Cache Poisoning

Usually a list of service providers is cached in the service requester. When a malicious node has a chance to compromise this cache by advertising fake services, then the service requester might always connect to this fake service provider. This attack is applicable to temporary public hotspot, home, enterprise, Mesh and 6LowPAN networks.

3.6. Harming Privacy

If a malicious node is in any subnet (WLAN and WAN) of a network, it can learn about all services available in this network. The DNS-SD discloses some critical information about resources in this network which might be harmful to privacy. This attack is applicable to temporary public hotspot and enterprise networks.

3.7. Resource spoofing

Resource owners in the network have permission to have the same name for load balancing. A malicious node can claim to be one of the load balanced resource devices and maliciously respond to requests. This is applicable to temporary public hotspot and enterprise networks.

3.8. Dual stack attacks

Having both IPv4 and IPv6 in the same network and trying to aggregate service discovery traffic on both IP stacks might cause new security flaws during the conversion or aggregation of this traffic. It can be similar to what explained here as an aggregated traffic or lead to a wide range of spoofing attacks. This attack is applicable to home, enterprise and temporary public hotspots.

3.9. MAC address spoofing

In a wireless environment where MAC address filtering is in use to avoid any malicious node joining to the network, a malicious node can easily spoof the MAC address of a legitimate node and join the network and perform malicious activities. This attack is applicable to temporary public networks and enterprise networks.

3.10. Privacy Protection Mechanisms

3.10.1. The Use of Random Data

Using a random name for services or devices or the use of random

numbers wherever possible, might prevent exposing the exact model or exact information regarding the DNS-SD service providers (e.g. printers, etc.) in the network to the attackers. However, this approach cannot be used for some standard information that the protocol needs to carry in order to offer service to other nodes. Otherwise, this random information was exchanged and agreed on between service providers and service requesters beforehand. This is exactly against the nature of zero conf protocols, i.e., DNS-SD

3.10.2. Data Encryption

Encrypting the whole DNS-SD message is another way to hide the critical information in the network. But this approach might not fit well to the nature of this protocol. The reason is because these devices usually respond to anonymous service discovery requests. So, the attacker can also submit and request the same information. In other words, encryption in this stage is only extra efforts without having any benefit from it.

3.11. Authorization of a Service Requester

3.11.1. The Use of an Access List

There can be an access list on each service providers with the list of IP addresses that can use these services. Then the service providers can use mechanisms to authorize the service requesters or to securely authenticate them with minimum interaction (zero configuration). This approach prevents the service providers from unauthorized use by an attacker. There are currently some mechanisms available -- SAVI-DHCP, CGA-TSIG, etc.

3.11.1.1. SAVI-DHCP

SAVI-DHCP [DHCP-SAVI] approach uses a simple mechanism in switches or devices that knows information about the ports of switches to filter any malicious traffic. This mitigates attacks on DHCP server spoofing and can make sure that nobody can spoof the IP address of the service providers.

3.11.1.2. CGA-TSIG

CGA-TSIG [cga-tsig] is another possible solution that can provide the node with secure authentication, data integrity and data confidentiality. It provides the node with zero or minimal configuration and prevents IP spoofing. This is useful when the node needs to update any record on an unicast DNS or there is an access list on service providers. This approach can be used to authenticate and authorize a node to use a service or a device.

3.11.1.3. DNS over DTLS

3.11.2. The Use of Shared Secret

A shared secret (e.g. a password) can be shared among the service requesters. Then this value can be used to access the service providers and authenticated on them. However, this approach has a disadvantage when one of the nodes in this network that carries this shared secret is compromised then the attacker can also have unauthorized access to these services. Sharing and re-sharing this shared secret does not fit to the zero conf nature of DNS-SD protocol.

3.12. Authorization of a Service Provider

It is really important for the service requesters to ensure that the one claim to be a service provider (e.g. a printer) is really a service provider and its identity has not been forged by the attacker. The service requester needs to receive the IP address of service providers in a secure manner. There are some approaches that can be used for this purpose such as SAVI-DHCP, Router Advertisement. There are also some mechanisms that can be used in service requesters to complete this authentication and authorization processes such as CGA-TSIG, DNS over TLS

3.12.1. SAVI-DHCP

The DHCP server can carry this information and send it to the service requesters at the same time as the service requesters receive a new IP address from the DHCP servers.

3.12.2. Router advertisement

If Neighbor Discovery Protocol (NDP) [RFC4861] or Secure Neighbor Discovery (SeND) [RFC3971] are in use, then an option can be added to a router advertisement message which carries required information regarding the IP addresses of service providers. This is especially secure when SeND is in use.

3.13. Other Security Considerations

Since a WLAN might also cover a part of city, it is really important to make sure that there is required filtering in edge networks to avoid distribution of mDNS/DNS-SD messages beyond the enterprise networks.

3.14. Not Usable Security Mechanisms

There are some other security mechanisms that are not fit to the zero conf nature of DNS-SD protocol but might be useable in future.

3.14.1. DNSSEC

Due to the pre-configuration required for DNSSEC on each nodes and DNS servers, it is not an ideal solution mechanism for zero config services. It might also necessary to access to internet to verify the DNSSEC keys and prevent IP spoofing (ask the trusted anchors the validity of the DNSSEC keys)

3.14.2. IPsec

IPsec is another security protection mechanism. Similar to DNSSEC, it requires manual step for the configuration of the nodes. However, recently there are some new drafts to automate this process. This is, of course, might not be an ideal solution for DNS-SD. This is because as explained in section 4.1.2 encryption of the whole message might not be really helpful since the attacker can also request the same service.

4. Security Considerations

This document documents the security of mDNS and DNS-SD. It does not introduce any additional security considerations

5. IANA Considerations

There is no IANA consideration

6. Acknowledgements

The author would like to thank all those people who directly helped in improving this draft, especially John C. Klensin, Douglas Otis and Dan York

7. References

7.1. Normative References

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

[RFC6762] Cheshire, S., Krochmal, M.,"Multicast DNS", RFC 6762, February 2013

[RFC6763] Cheshire, S., Krochmal, M., "DNS-Based Service Discovery", RFC 6763, February 2013

[RFC6275] Perkins, C., Johnson, D., Arkko, J., "Mobility Support in IPv6", RFC 6275, July 2011

[RFC3833] Atkins, D., Austein, R., "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004

[RFC3971] Arkko, J., Kempf, J., Zill, B., and Nikander, P., "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., Soliman, H., "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

7.2. Informative References

[requirement] Lynn, K., Cheshire, S., Blanchet, M., Migault, D., " Requirements for Scalable DNS-SD/mDNS Extensions", http://tools.ietf.org/html/draft-ietf-dnssd-requirements-04, October 2014

[DHCP-SAVI] Bi, J., Wu, J., Yao, G, Baker, F.,"SAVI Solution for DHCP", http://tools.ietf.org/html/draft-ietf-savi-dhcp-23, April 2014

[cga-tsig] Rafiee, H., Loewis, M., Meinel, C.,"Transaction SIGnature (TSIG) using CGA Algorithm in IPv6", http://tools.ietf.org/html/draft-rafiee-intarea-cga-tsig , June 2014

Authors' Addresses

Hosnieh Rafiee
HUAWEI TECHNOLOGIES Duesseldorf GmbH
Riesstrasse 25, 80992
Munich, Germany
Phone: +49 (0)162 204 74 58
Email: ietf@rozanak.com

IETF
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

A. Sullivan
Dyn

October 27, 2014

On Interoperation of Labels Between mDNS and DNS
draft-sullivan-dnssd-mdns-dns-interop-01

Abstract

Despite its name, DNS-Based Service Discovery can use naming systems other than the Domain Name System when looking for services. Different name systems use different conventions for the characters allowed in any name. In order for DNS-SD to be used effectively in environments where multiple different name systems are in use, it is important to attend to differences in the underlying technology. This memo presents an outline of the requirements for selection of labels for mDNS and DNS when they are expected to interoperate in this manner.

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 April 30, 2015.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Conventions and terms used in this document 3
2. Requirements for a profile for label interoperation 3
3. DNS-SD portions 4
3.1. The <Instance> Portion of the Service Instance Name . 4
3.2. The <Service> Portion of the Service Instance Name . 5
3.3. The <Domain> Portion of the Service Instance Name . . 5
4. Acknowledgements 6
5. IANA Considerations 6
6. Security Considerations 6
7. Informative References 6
Author's Address 7

1. Introduction

DNS-Based Service Discovery (DNS-SD, [RFC6763]) specifies a mechanism for discovering services using queries both to the Domain Name System (DNS, [RFC1034], [RFC1035]) and to Multicast DNS (mDNS, [RFC6762]). Conventional use of the DNS generally follows the host name rules [RFC0952] for labels -- the so-called LDH rule. That convention is the reason behind the development of Internationalized Domain Names for Applications (IDNA2008, [RFC5890], [RFC5891], [RFC5892], [RFC5893], [RFC5894], [RFC5895]). It is worth noting that the LDH rule is a convention, and not a strict rule of the DNS. It is assumed to be true widely enough, however, that in many circumstances names cannot be used unless they cleave to the LDH rule.

At the same time, mDNS requires that labels be encoded in UTF-8, and permits a range of characters in labels that are not permitted by IDNA2008 or the LDH rule. For example, mDNS encourages the use of spaces and punctuation in mDNS names (see [RFC6763], section 4.1.3). It does not restrict which Unicode code points may be used in those labels, so long as the code points are UTF-8 in Net-Unicode [RFC5198] format.

Users of applications are, of course, frequently unconcerned with (not to say oblivious to) the name-resolution system(s) in service at any given moment, and are inclined simply to use the same names in different contexts. As a result, the same string might be tried as a name using different name resolution technologies. If DNS-SD is to be used in an environment where both mDNS and DNS are to be queried

for services, then some parts of the names to be queried will need to be compatible with the rules and conventions for both DNS and mDNS.

One approach to interoperability under these circumstances is to use a single operational convention for names under the different naming systems. This memo assumes such a use profile, and outlines what is necessary to make it work.

It is worth noting that users of DNS-SD do not use the service discovery names in the same way that users of other domain names might. Most domain names might as easily be typed in as direct user input as any other method. But the service discovery context generally assumes users are picking a service from a list. As a result, the sorts of application considerations that are appropriate to the general-purpose DNS name, and that resulted in the A-label/ U-label (see below) in IDNA2008, are not the right approach for DNS-SD.

1.1. Conventions and terms used in this document

Wherever appropriate, this memo uses the terminology defined in Section 2 of [RFC5890]. In particular, the reader is assumed to be familiar with the terms "U-label", "LDH label", and "A-label" from that document. Similarly, the reader is assumed to be familiar with the U+NNNN notation for Unicode code points used in [RFC5890] and other documents dealing with Unicode code points. In the interests of brevity and consistency, the definitions are not repeated here.

This memo refers to names in the DNS as though the LDH rule and IDNA2008 are strict requirements. They are not. DNS labels are, in principle, just collections of octets, and therefore in principle the LDH rule is not a constraint. In practice, applications often intercept labels that do not conform to the LDH rule and apply IDNA and other transformations.

The term "owner name" (common to the DNS vernacular) is used here to apply not just to the names to be looked up in the DNS, but to any name that might be looked up either in the DNS or using mDNS.

2. Requirements for a profile for label interoperation

Any interoperability between mDNS and DNS will require interoperability across some of the portions of a DNS-SD Service Instance Name (see Section 3) that are implicated in regular mDNS and DNS lookups. Only some portions are implicated. In any case, if a given portion is implicated, the profile will need to apply to all labels in that portion.

In addition, because DNS-SD Service Instance Names can be used in a domain name slot, care must be taken by DNS-SD resolvers to undertake the special processing outlined here, so that DNS-SD portions that do not use IDNA2008 will not be treated as U-labels and will not undergo IDNA processing.

Because the profile will need to apply to names that might need to interoperate with names in the DNS, and because mDNS permits labels that IDNA does not, the profile might reduce the labels that could be used with mDNS. Consequently, some recommendations from [RFC6763] will not really be possible to implement using names subject to the profile. In particular, [RFC6763], section 4.1.3 recommends that labels always be stored and communicated as UTF-8, even in the DNS. Because IDNA2008 libraries will treat any Unicode-encoded labels as candidate U-labels and attempt to perform resolution in A-label form, the advice to store and transmit labels as UTF-8 in the DNS is likely to encounter problems. In particular, the <Domain> part of a Service Instance Name is unlikely to be found in its UTF-8 form in the public DNS tree for zones that are using IDNA2008. By contrast, mDNS normally uses UTF-8.

U-labels cannot contain upper case letters. That restriction extends to ASCII-range upper case letters that work fine in LDH-labels. It may be confusing that the character "A" works in the DNS when none of the characters in the label has a diacritic, but does not work when there is such a diacritic in the label. Labels in mDNS names may contain upper case characters, so the profile will need either to restrict the use of upper case or come up with a reliable and predictable (to users) convention for case folding even in the presence of diacritics.

3. DNS-SD portions

DNS-SD specifies three portions of the owner name for a DNS-SD resource record. These are the <Instance> portion, the <Service> portion, and the <Domain>. The owner name made of these three parts is called the Service Instance Name. It is worth observing that a portion may be more than one label long. See [RFC6763], section 4.1.

3.1. The <Instance> Portion of the Service Instance Name

[RFC6763] is clear that the <Instance> portion of the Service Instance Name is intended for presentation to users, and therefore virtually any character is permitted in it. There are two ways that a profile might address this portion.

The first way would be to treat this portion as likely to be intercepted by system-wide IDNA-aware resolvers. In this case, the

portion needs to be made subject to the profile, thereby curtailing what characters may appear in this portion. This approach permits DNS-SD to use any standard system resolver but presents inconsistencies with the DNS-SD specification and with DNS-SD that is exclusively mDNS-based. Therefore, this strategy is rejected.

Instead, DNS-SD implementations can intercept the <Instance> portion of a Service Instance Name and ensure that those labels are never handed to IDNA-aware resolvers that might attempt to convert these labels into A-labels. Under this approach, the DNS-SD <Instance> portion works as it always does, but at the cost of using special resolution code built into the DNS-SD system.

3.2. The <Service> Portion of the Service Instance Name

DNS-SD includes a <Service> component in the Service Instance Name. This component is not really user-facing data, but is instead control data embedded in the Service Instance Name. This component includes so-called "underscore labels", which are labels prepended with U+005F (_). The underscore label convention was established by DNS SRV ([RFC2782]) for identifying metadata inside DNS names. A system-wide resolver (or DNS middlebox) that cannot handle underscore labels will not work with DNS-SD at all, so it is safe to suppose that such resolvers will not attempt to do special processing on these labels. Therefore, the <Service> portion of the Service Instance Name will not be subject to the profile.

3.3. The <Domain> Portion of the Service Instance Name

The <Domain> portion of the Service Instance Name forms an integral part of the QNAME submitted for DNS resolution, and a system-wide resolver that is IDNA2008-aware is likely to interpret labels with UTF-8 in the QNAME as candidates for IDNA2008 processing. Operators of Internationalized Domain Names will almost certainly publish them in the DNS as A-labels. Therefore, these labels will need to be subject to the profile. DNS-SD implementations ought to identify the <Domain> portion of the Service Instance Name and treat it subject to IDNA2008 in case the domain is to be queried from the global DNS. This is different to the rule for resolution published in [RFC6763].

One might argue against this restriction on either of two grounds:

1. It is possible the names may be in the DNS in UTF-8, and RFC 6763 already specifies a fallback strategy of progressively attempting first the U-label lookup and then the A-label lookup.

2. Zone administrators that wish to support DNS-SD can publish a UTF-8 version of the zone along side the A-label version of the zone.

The first of these is rejected because it represents a potentially significant increase in DNS lookup traffic for no value. It is possible for a DNS-SD application to identify the <Domain> portion of the Service Instance Name. The standard way to publish IDNs on the Internet uses IDNA. Therefore, additional lookups should not be encouraged. When [RFC6763], the bulk of IDNs were lower in the tree, but now that there are internationalized labels in the root zone, it seems reasonable to use only the single lookup strategy.

The second reason depends on the idea that it is possible to maintain two names in sync with one another. This is not strictly speaking true, although in this case the domain operator could simply create a DNAME record [RFC6672] from the UTF-8 name to the IDNA2008 zone. This still, however, relies on being able to reach the (UTF-8) name in question, and it is unlikely that the UTF-8 version of the zone will be delegated from anywhere. Moreover, in many organizations the support for DNS-SD and the support for domain name delegations are not performed by the same department, and depending on a co-ordination between the two will make the system more fragile or slower or both.

4. Acknowledgements

The author gratefully acknowledges the insights of Stuart Cheshire and Kerry Lynn.

5. IANA Considerations

This memo makes no requests of IANA.

6. Security Considerations

This memo presents some requirements for future development, but does not specify anything. Therefore, it has no implications for security.

7. Informative References

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.

[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.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

Author’s Address

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

Email: asullivan@dyn.com