Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2016

C. Huitema
Microsoft

March 9, 2016

Privacy Extensions for DNS-SD
draft-huitema-dnssd-privacy-00.txt

Abstract

DNS-SD allows discovery of services published in DNS or MDNS. The publication normally disclose information about the device publishing the services. There are use cases where devices want to communicate without disclosing their identity, for example two mobile devices visiting the same hotspot. We propose a method to obfuscate the identification information published by DNS-SD.

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 September 10, 2016.

Table of Contents

1. Introduction

There are cases when nodes connected to a network want to provide or consume services without exposing their identity to the other parties connected to the same network. Consider for example a traveller wanting to upload pictures from a phone to a laptop when connected to the Wi-Fi network of an Internet cafe, or two travellers who want to share files between their laptops when waiting for their plane in an airport lounge.

We expect that these exchanges will start with a discovery procedure using DNS-SD [RFC6763]. One of the devices will publish the availability of a service, such as a picture library or a file store in our examples. The user of the other device will discover this service, and then connect to it.

When analysing these scenarios in Section 2, we find that the DNS-SD messages leak identifying information such as instance name, host name or service properties. We review the design constraint of a solution in Section 3, and describe the proposed solution in Section 4.

1.1. Requirements

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 [RFC2119].

2. Privacy implications of DNS-SD

DNS-Based Service Discovery (DNS-SD) is defined in [RFC6763]. It allows nodes to publish the availibility of an instance of a service by inserting specific records in the DNS ([RFC1033], [RFC1034], [RFC1035]) or by publishing these records locally using multicast DNS (MDNS) [RFC6762]. The service availability will be described in three types of records:

PTR Record: Associate the service name in the domain with the "instance" name published by the node.

SRV Record: Provides the node name, port number, priority and weight associated with the service instance, in conformance with [RFC2782].

TXT Record: Provides a set of attribute-value pairs describing specific properties of the service instance.

In the remaining subsections, we will review the privacy issues related to publishing instance names, node names, service attributes and other data, as well as review the implications of using the discovery service as a client.

2.1. Privacy implication of publishing instance names

In the first phase of discovery, the client will obtain a copy of all the PTR records associated to a service in a given naming domain. Each record contains a domain name starting with an instance name. Instance names are free form description of the instance, and are meant to convey enough information so discovery clients can easily select the desired service. Section 4 of [RFC6763] give the following example for the instance names of a printer service:

```
Building 2, 1st Floor  .  example  .  com  .
Building 2, 2nd Floor  .  example  .  com  .
Building 2, 3rd Floor  .  example  .  com  .
Building 2, 4th Floor  .  example  .  com  .
```

Nodes that use DNS-SD in a mobile environment will rely on the specificity of the instance name to identify the desired service. In our example of users wanting to upload pictures to a laptop in an Internet Cafe, the list of available services may look like:

```
Alice's notebook       .  local  .
Bob's laptop           .  local  .
Image store for Carol  .  local  .
```

Alice will see the list on her phone and understand intuitively that she should pick the fist item. The discovery will "just work." It will also reveal to anybody who cares that Alice is currently visiting the Internet Cafe.

2.2. Privacy implication of publishing node names

The SRV records contain the DNS name of the node publishing the service. Typical implementations construct this DNS name by concatenating the "host name" of the node with the name of the local domain. The privacy implications of this practice are reviewed in [I-D.ietf-intarea-hostname-practice]. Depending on naming practices, the host name is either a strong identifier of the device, or at a minimum a partial identifier. It enables tracking of the device, and by extension of the device's owner.

2.3. Privacy implication of publishing service attributes

The TXT records contain a set of attribute and value pairs characteristics of the service implementation. These attributes reveal some information about the devices that publishes the service. The amount of information will vary widely with the particular service and its implementation:

- Some attributeslike the paper size available in a printer, are the same on many devices, and thus only provides limited information to a tracker.

- Attributes that have freeform values, such as the name of a directory, may reveal much more information.

Combinations of attributes have more information power than specific attributes, and can potentially be used for "fingerprinting" a specific device.

2.4. Device fingerprinting

The combination of information published in DNS-SD has the potential to provide a "fingerprint" of a specific device. Such information includes:

- The list of services published by the device, which can be retrieved because the SRV records will point to the same host name.
- The specific attributes describing these services.
- The port numbers used by the services.
- The values of the priority and weight attributes in the SRV records.

This combination of services and attribute will often be sufficient to identify the version of the software running on a device. If a device publishes many services with rich sets of attributes, the combination may be sufficient to identify the specific device.

2.5. Privacy implication of discovering services

The consumers of services engage in discovery, and in doing so do reveal some information such as the list of services that they are interested in and the domains in which they are looking for the services. When the clients select specific instances of services, they reveal their preference for these instances.

In first analysis, the leakage of information by lients looks benign compared to the disclosures made by the servers. There may be a concern when the client is attempting to use rare services.

3. Design of DNS-SD privacy mitigations

Ah Ah.

3.1. Obfuscated instance names

The privacy issues described in Section 2.1 can be solved by obfuscating the instance names. Instead of a user friendly description of the instance, the nodes will publish a random looking string of characters. To prevent tracking over time and location, different string values should be used at different locations, or at different times.

Authorized parties should be able to "de-obfuscate" the names, while non-authorized third parties will not be. For example, if both Alice notebook and Bob's laptop use an obfuscation process, the list of available services should appear differently to them and to thrid parties. Alice's phone will be able to de-obfuscate the name of Alice's notebook, but not that of Bob's laptop. Bob's phone will do the opposite. Carol will do neither.

Alice will see something like:

```
GobbeldygookBlaBlaBla (Alice's notebook) .  local  .
Abracadabragooklybok                     .  local  .
Image store for Carol                    .  local  .
```

Bob will see:

```
GobbeldygookBlaBlaBla                  .  local  .
Abracadabragooklybok (Bob's laptop)    .  local  .
Image store for Carol                  .  local  .
```

Carol will see:

```
GobbeldygookBlaBlaBla  .  local  .
Abracadabragooklybok   .  local  .
Image store for Carol  .  local  .
```

In that example, Alice, Bob and Carol will be able to select the appropriate instance. It would probably be preferable to filter out the obfuscated instance names, to avoid confusing the user. In our example, Alice and Bob have updated their software to understand obfuscation, and they could easily filter out the obfuscated strings that they do not like. But Carol is not using this system, and we could argue that her experience is suboptimal.

The suboptimal experience with unmodified software could be avoided if the obfuscated service records were published using different service names, or using different domain names. This would of course make management a bit more complex, and is thus debatable.

3.2. Randomized host names

Instead of publishing their actual name in the SRV records, nodes could publish a randomized name. That the solution argued for in [I-D.ietf-intarea-hostname-practice].

Randomized host names will prevent some of the tracking. Host names are typically not visible by the users, and randomizing host names will probably not cause much usability issues.

3.3. Timing of obfuscation and randomization

It is important that obfuscation of instance names be performed at the right time, and that the obfuscated names change in synchrony with other identifiers, such as MAC Addresses, IP Addresses or host names. If the randomized host name changed but the instance name remained constant, an adversary would have no difficulty linking the old and new host names. Similarly, if IP or MAC addresses changed but host names remained constant, the adversary could link the new addresses to the old ones using the published name.

The problem is handled in [I-D.ietf-intarea-hostname-practice], which recommends to pick a new random host name at the time of connecting to a new network. The instance names should be obfuscated at the same time, or maybe use the randomized host name as input in the randomization process.

3.4. Fingerprint resistance

Difficult...

3.5. A note on Private DNS services

The DNS Private Exchange working group develops mechanisms to provide confidentiality to DNS transactions, addressing the problems outlined in [RFC7626]. The solutions being developed include DNS over TLS [I-D.ietf-dprive-dns-over-tls] and DNS over DTLS [I-D.ietf-dprive-dnsodtls].

We could imagine that DNS-SD nodes are configure to update and retrieve DNS records using DNS over TLS or DNS over DTLS, but a number of problems can arise:

- Discovery queries are scoped by the domain name within which services are published. As nodes move and visit arbitrary networks, there is no guarantee that the domain services for these networks will be accessible using DNS over TLS or DNS over DTLS.

- Information placed in the DNS is considered public. Even if the server does support DNS over TLS, third parties will still be able to discover the content of PTR, SRV and TXT records.

- Neither DNS over TLS nor DNS over DTLS applies to MDNS.

In short, DNS ovr TLS and DNS over DTLS solve a different problem, and are not a solution for DNS-SD privacy.

4. Privacy extensions for DNS-SD

The proposed solution uses the following components:

- The host names are randomized to prevent tracking.
- Nodes provide an Instance Discovery Key to other nodes authorized to discover the service instance,
- The Instance Discovery Key is combined with a random seed to obfuscate the instance names,
- Nodes engaged in discovery attempt to de-obfuscate the instance names using the set of Instance Discovery Key that they know about,

These components are detailed in the following subsections.

4.1. Randomized Host Name

Nodes publishing services with DNS-SD and concerned about their privacy MUST use a randomized host name. The randomized name MUST be changed when network conectivity changes, to avid the correlation issues described in Section 3.3. The randomized host name MUST be used in the SRV records describing the service instance, and the corresponding A or AAAA records MUST be made available through DNS or MDNS, within the same scope as the PTR, SRV and TXT records used by DNS-SD.

If the link-layer address of the network connection is properly obfuscated (e.g. using MAC Address Randomization), The Randomized Host Name MAY be computed using the algorithm described in section 3.7 of [I-D.ietf-dhc-anonymity-profile]. If this is not possible, the randomized host name SHOULD be constructed by simply picking a 48 bit random number meeting the Randomness Requirements for Security expressed in [RFC4075], and then use the hexadecimal representation of this number as the obfuscated host name.

4.2. Instance Discovery Key

The obfuscation and de-obfuscation of instance names is controlled by the Instance Discovery Key. Each device publishing a service instance configures an Instance Discovery Key associated with the service instance.

The Instance Key SHOULD be at least 16 bytes long (128 bits). Its content SHOULD meet the Randomness Requirements for Security expressed in [RFC4075].

4.3. Composing Obfuscated Instance Names

The obfuscated instance name is composed of two components, a seed and a hash, encoded in BASE64 ([RFC2045] section 6.8) and separated by a dot:

```
instance_name = <base64_seed> "." <base64_hash>
```

The seed is derived algorithmically from the randomized host name. If the randomized name changes, new instance names SHOULD be computed and the corresponding records SHOULD be published in order to meet the requirement defined in Section 3.3.

The complete instance name MUST be genrated using the following process:

```
long_seed = HASH(randomized_host_name)
seed = first 12 bytes of long_seed
long_hash = HASH(seed | instance_discovery_key )
instance_hash = first 12 bytes of long_hash
instance_name = BASE64(seed) "." BASE64(instance_hash)
```

In this formula, HASH SHOULD be the function SHA256 defined in [RFC4055], unless otherwise specified. Implementers MAY eventually replace SHA256 with a stronger algorithm.

The algorithm produces seeds and hash that are encoded as 16 BASE64 characters. The resulting instance name is 33 characters long, which fits within the 63 characters limit defined in [RFC6763].

4.4. De-Obfuscation of Instance Names

De-obfuscation of instance names assumes that authorized nodes are provisioned with three elements for each discoverable instance:

- the de-obfuscated instance name,
- a copy of the instance_discovery_key,
- optionally, the identifier of the HASH function used by the publisher.

A given node may be provisioned do discover many instances. For example, Alice's phone may know about Alice's laptop and Alice's desktop. It might also know of Bob's laptop, if Alice and Bob have agreed to share such information.

To de-obfuscate the instance names, nodes performing discovery should obtain the list of PTR records published for the service and domain being searched and then do the following:

- Test whether the instance name contains the base64 encoding of a seed and hash as defined in Section 4.3. If it is not in that form, the name is not considered obfuscated.

- Retrieve the binary seed and hash from the base64 encoding.

- For each known instance discovery key, compute whether the hash of the seed and key, and compare it to the published hash.

- If there is a hash, the de-obfuscated name of the instance is the de-obfuscated name associated with the matching instance discovery key

5. Security Considerations

This document specifies a method to protect the privacy of service publishing nodes. This is especially useful when operating in a public space. Obfuscating the identity of the publishing nodes prevents some forms of "targeting" of high value nodes.

Obfuscating the identity of the publishing nodes does not provide any form of access control. It will not prevent attackers from trying to access the services.

The cost of the de-obfuscation algorithm scales as the product of the number of authorized publishers known by the client, times the number of obfuscated services published in the searched name domain. Attackers could potentially publish a large number of bogus instances of a service, forcing a high computation cost on discovery clients. While this potential denial of service attack is concerning, we note that this is merely an aggravation of a flooding attacks against DNS-SD.

6. IANA Considerations

This draft does not require any IANA action.

7. Acknowledgments

This draft results from initial discussions with Dave Thaler.

8. References

8.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <http://www.rfc-editor.org/info/rfc2045>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, DOI 10.17487/RFC4055, June 2005, <http://www.rfc-editor.org/info/rfc4055>.

[RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, DOI 10.17487/RFC4075, May 2005, <http://www.rfc-editor.org/info/rfc4075>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <http://www.rfc-editor.org/info/rfc6763>.

8.2. Informative References

[I-D.ietf-dhc-anonymity-profile] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity profile for DHCP clients", draft-ietf-dhc-anonymity-profile-08 (work in progress), February 2016.

[I-D.ietf-dprive-dns-over-tls] Zi, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over TLS", draft-ietf-dprive-dns-over-tls-07 (work in progress), March 2016.

[I-D.ietf-dprive-dnsodtls] Reddy, T., Wing, D., and P. Patil, "DNS over DTLS (DNSoD)", draft-ietf-dprive-dnsodtls-04 (work in progress), January 2016.

[I-D.ietf-intarea-hostname-practice]
Huitema, C. and D. Thaler, "Current Hostname Practice Considered Harmful", draft-ietf-intarea-hostname-practice-00 (work in progress), October 2015.

[RFC1033] Lottor, M., "Domain Administrators Operations Guide", RFC 1033, DOI 10.17487/RFC1033, November 1987, <http://www.rfc-editor.org/info/rfc1033>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <http://www.rfc-editor.org/info/rfc2782>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <http://www.rfc-editor.org/info/rfc6762>.

[RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <http://www.rfc-editor.org/info/rfc7626>.

Author's Address

Christian Huitema
Microsoft
Redmond, WA 98052
U.S.A.

Email: huitema@microsoft.com