Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

T. Lemon
Nominum, Inc.

March 21, 2016

Homenet Naming and Service Discovery Architecture
draft-lemon-homenet-naming-architecture-00

Abstract

This document recommends a naming and service discovery resolution architecture for homenets. This architecture covers the publication and resolution of names of hosts on the homenet both within the homenet and on the public internet, and the use of such names for offering and discovering services that exist on the homenet both within the homenet and on the public internet. Security and privacy implications and techniques for automatically and administratively setting security and privacy policies for such names are also described.

Table of Contents

1. Introduction

Associating domain names with hosts on the Internet is a key factor in enabling communication with hosts, particularly service discovery. In order to provide name service, several provisioning mechanisms must be available:

- Provisioning of a namespace in which names can be published and services advertised
- Associating a name within that namespace to the set of IP addresses on which a host is reachable
- Advertising services available on the local network and associating those services with names published in the namespace
- Distribution of names published in that namespace to servers that can be queried in order to resolve names
- Correct advertisement of name servers that can be queried in order to resolve names
- Timely removal of published names when they are no longer in use

Homenet adds the following considerations:

1. Some names may be published in a broader scope than others. For example, it may be desirable to advertise some homenet services to users who are not connected to the homenet. However, it is unlikely that all services published on the home network would be appropriate to publish outside of the home network. In many cases, no services will be appropriate to publish outside of the network, but the ability to do so is required.

2. Users cannot be assumed to be skilled or knowledgeable in name service operation, or even to have any sort of mental model of how these functions work. With the possible exception of policy decisions, all of the operations mentioned here must reliably function automatically, without any user intervention or debugging. Even to the extent that users may provide input on policy, such as whether a service should or should not be advertised outside of the home, the user must be able to safely provide such input without having a correct mental model of how naming and service discovery work, and without being able to reason about security in a nuanced way.

3. Because user intervention cannot be required, naming conflicts must be resolved automatically, and, to the extent possible, transparently.

4. Where services are advertised both on and off the home network, differences in naming conventions that may vary depending on the user's location must likewise be transparent to the end user.

5. Hosts that do not implement any homenet-specific capabilities must still be able to discover and access services on the homenet, to the extent possible.

6. Homenet explicitly supports multihoming--connecting to more than one Internet Service Provider--and therefore support for multiple provisioning domains [6] is required to deal with situations where the DNS may give a different answer depending on whether caching resolvers at one ISP or another are queried.

7. Multihomed homenets may treat all service provider links as equivalent, or may treat some links as primary and some as backup, either because of differing transit costs or differing performance. Services advertised off-network may therefore be advertised for some links and not others.

In addition to these considerations, there may be a need to provide for secure communication between end users and the user interface of the home network, as well as to provide secure name validation (e.g., DNSSEC). Secure communications require that the entity being secured have a name that is unique and can be cryptographically authenticated within the scope of use of all devices that must communicate with that entity. Because it is very likely that devices connecting to one homenet will be sufficiently portable that they may connect to many homenets, the scope of use must be assumed to be global. Therefore, each homenet must have a globally unique name.

1.1. Existing solutions

Previous attempts to automate naming and service discovery in the context of a home network are able to function with varying degrees of success depending on the topology of the home network. For example, Multicast DNS [4] can provide naming and service discovery [5], but only within a single multicast domain.

The Domain Name System provides a hierarchical namespace [1], a mechanism for querying name servers to resolve names [2], a mechanism for updating namespaces by adding and removing names [3], and a mechanism for discovering services [5]. Unfortunately, DNS provides no mechanism for automatically provisioning new namespaces, and

secure updates to namespaces require pre-shared keys, which won't work for an unmanaged network. DHCP can be used to populate names in a DNS namespace; however at present DHCP cannot provision service discovery information.

Hybrid Multicast DNS [7] proposes a mechanism for solving the single-multicast-domain problem. However, it has serious shortcomings as a solution to the Homenet naming problem. The most obvious shortcoming is that it requires that every multicast domain have a separate name. This then requires that the homenet generate names for every multicast domain, and requires that the end user have a mental model of the topology of the network in order to guess on which link a given service may appear.

2. Homenet Naming Database

In order to resolve names, there must be a place where names are stored. There are two ways to go about this: either names are stored on the devices that own them, or they are stored in the network. This isn't a clean dichotomy, however: it's possible for the source of truth about a name to be owned by the device, while the resolution of the name is owned by a service separate from the device. Additionally, if names are owned by devices, conflicts can arise, since two devices might present the same name by default or by accident. Further, devices can be attached to more than one network, in which case we want the same name to identify them on both networks. Additionally, although homenets are self-configuring, user customization is permitted and useful.

In order to achieve this, the Homenet Naming Database (HNDB) provides a persistent central store into which names can be registered

2.1. Global Name

Every homenet must be able to have a name in the global DNS hierarchy which serves as the root of the zone in which the homenet publishes its public namespaces. Homenets that do not yet have a name in the global namespace use the homenet special-use TLD [TBD1] as their "global name" until they are configured with a global name.

[It's tempting to have the homenet generate a UUID-like name that can be used as a global name, but we really don't want that, because it will be quite ugly to the user, difficult to remember, and therefore not protective against the kinds of mistakes we'd want such a name to protect against. It's better as a security UI to have the user see a name that is the same on all homenets. This will allow the user to fairly easily notice that they see the same name on every homenet. To present a name that isn't really unique and isn't easily

identified as anything other than "random gibberish," may lull the end user into a sense that they are talking to the right homenet when they see the random gibberish, without realizing that actually it's different gibberish and they aren't talking to their own homenet.]

A homenet's global name can be a name that the homenet user has registered on their own in the DNS using a public DNS registrar. However, this is not required and, indeed, presents some operational challenges. It can also be a name under a domain owned by one of the user's service providers, or managed by some service provider that specifically provides homenet naming services.

For most end-users, the second or third options will be preferable. It will allow them to choose an easily-remembered homenet domain name under an easily-remembered service provider subdomain, and will not require them to maintain a DNS registration.

[This doesn't belong in the arch document, at least not in this part, but I'm writing it down because I think we should discuss it and figure out exactly what we should recommend, and I do believe we should recommend something here and not just hope for the best. I am also hoping to stave off frivolous, privacy-harming patents by publishing this now:]

Providers of either of these types of homenet naming service should offer a selection of different provider TLDs rather than a flat namespace, so as to avoid the exampleuser1997 problem. So for example rather than having a single namespace "isp.example.com", the provider should have a series of subdomains, like "grapefruit.example.com", "warrior.example.com", "koala.example.com", "rocket.example.com", and so forth.

Some small number of such subdomains should be presented to the user when registering. Subdomains with more than perhaps 50,000 homenets registered in them should never be presented for registration, to avoid chosen name collisions. If the user is unable to find a namespace they like, it may be beneficial to allow the user to cycle through a larger set of namespaces. The user should wind up with a global name like "hamburger.warrior.example.com".

This specification may seem frivolous or overly-prescriptive. The reason for being this specific is that it is important for the user to be able to quickly choose a memorable name that doesn't contain personal identifying information. Getting this user interface right has significant implications for the user's privacy and security. Any user interface that meets the criterion that the user can quickly choose a memorable name that doesn't contain personal information will address this requirement. The user should be specifically told

not to use personal information like birthdays or names of friends or pets, and should be encouraged to write the chosen name down until they have it memorized.

2.2. Local namespaces

Every homenet has two non-hierarchical local namespaces, one for associating DNS RRs with names, and one for associating DNS RRs with IP addresses. These namespaces are key-data stores, where for the name mapping the primary key is a single DNS label, and for the IP address mapping the primary key is an IP address. The secondary key is an RRtype, so that there can be more than one RRset per IP address, and each data element is an RRset of the corresponding RRtype.

Each RRset in each local namespace is marked with a flag that indicates whether it is to be public or private. Each RRset is also marked with a flag that indicates whether its availability is critical or best-effort. This flag only has meaning if the RRset is marked public.

Each RR that contains a name (e.g, a CNAME or SRV record) either contains a name in the local namespace or a global name. All global names are references to external services, not to services on the homenet. All local names are qualified with the homenet-specific special-use TLD, [TBD1].

[Question: do we need RRset granularity for these flags?]

The local namespace is maintained as a distributed database with copies on every homenet router. No copy is the master copy. Although the local namespace is non-hierarchical, it is permissible for it to contain RRtypes that contain delegations. However, from an operational perspective is is most likely better for the local namespace to be at the bottom of the delegation hierarchy, and so we do not recommend the use of such delegations.

2.3. Public namespaces

Every homenet has two public namespaces. These are copies of the private namespaces with four modifications:

1. Names with no public RRsets are not included in the public namespace.

2. RRs that contain IP addresses in the homenet's ULA prefix are omitted.

3. By default, RRs that contain IPv4 addresses are omitted, because IPv4 doesn't support renumbering. However, there should be a whitelist of IPv4 addresses that may be published, so that if the end user has static IPv4 addresses, those can be published.

4. If an RRset is marked best-effort rather than critical, RRs containing IP addresses that match prefixes assigned by backup links are omitted.

[Are there RRtypes or classes of DNSSD records that we want to always omit?]

Because the public namespaces are copies of the private namespaces, replication is not necessary: each homenet router automatically produces public namespaces by deriving them from the private namespaces using the above rules. The public namespaces can be derived on demand, or maintained automatically as updates are made to the private namespaces.

[Security/privacy considerations: It can be argued that public namespaces provide a means for botnets to publish rendezvous information. However, in fact this isn't really true because if botnets did so, it would be very obvious, so would wind up being used as a means of tracking and suppressing them. It's probably better to encourage this mistake rather than trying to prevent it, since the former benefits white hats, and the latter limits functionality.]

[Presumably we want it to be possible for devices that are meant to be public servers to publish their names in the DNS, but how do we automatically determine that a device is "meant" to be public? This needs thought.]

2.4. Adding Names

Several mechanisms are available for updating the HNDB.

2.4.1. mDNS Snooping

Homenet snoops mDNS for device names. Homenet does not defend names. If two devices with the same name appear on a link, mDNS handles collision resolution. If two devices with the same name appear on different links, homenet deterministically generates names for both devices using the link-layer address of each device, plus the name that device claimed. If a device that appears on two links is the same device (presents the same link-layer address or DHCID) then it is treated as a single device, with a single name. This whole discussion probably belongs in a separate draft.

2.4.2. DHCP DNS Update (stateful or stateless)

A and PTR records can be set up this way. Doesn't work (yet) for service discovery. Should we write a draft, or just rely on:

2.4.3. DNS Update

DNS updates can be send to any resolver in the homenet to add names to the local zone. If there is no conflict, the name is added; otherwise the update is rejected.

[Really, what we ought to do is just allow devices to declare an ID that is a public key, and then do DNS updates to the local service zone signed with the corresponding private key. Devices would also sign their mDNS claims with the same key, so that mDNS updates for devices that support this functionality can be ignored by the mDNS snooping agent.]

[Records added to the namespace that contain names are problematic: the hostname label is obvious, but what about the domain name? I think the answer is that these names shouldn't be qualified, and the name server should qualify them appropriately. What does mDNS do?]

[Add something about [TBD1] versus global name. If there is a global name, that is what we use for name resolution on the local network. This is necessary because of the security implications, both for DNSSEC and for PKI.]

2.5. Removing Names

mDNS: names time out

DHCP: names go away when lease expires, or, for stateless, when refresh timer expires

DNS Update: names have to have a lifetime, determined by the DNS server, and DNSSD devices that do DNS Update have to keep sending updates at appropriate intervals to reset the timer. If the lifetime of a name expires, the name is deleted. Names can also be deleted by a new update, which must either be signed with SIG(0) using a key published on the name, or else must come from an IP address published in the name if no key is published on the name (what if there's no IP address?)

2.6. Name Collisions

Covered under adding names. Say more?

2.7. Recovery from loss

In principle the names in the zone aren't precious. If there are multiple HNRs and one is replaced, the replacement recovers by copying the local namespaces and other info from the others. If all are lost, there are a few pieces of persistent data that need to be recovered:

- The global name
- The ZSK for both local namespaces
- Names configured statically through the UI

All other names were acquired dynamically, and recovery is simply a matter of waiting for the device to re-announce its name. We should have a protocol for informing devices that they should do this, either using an ND option or a DHCP message, or else devices should know to do a fresh announcement whenever the link goes away (but that doesn't work if the device is connected to the nearest router through a separate switch, so ND option is better).

There should be some way (ideally) for the global name to be recovered. How? This will cause problems with DNSSEC, because the private ZSK is lost, and we don't expect the user to be smart about keys. ZSK or KSK could be stored encrypted at the SP. Subject to brute force attacks if so, probably not a good idea. Better to just have a flag day? One answer is the end-user management RESTful API: if the end-user has a phone, the ZSK and other static info can be maintained in an app on the phone; this app can then be in touch with the homenet, and if the homenet finds that it is amnesiac, the app can notify the user. Of course, this is a potential attack--we don't want some other network the phone connects to to be able to steal the ZSK just by telling the app it's amnesiac.

2.8. Persistence

When the whole homenet goes away, can we recover the zone? Step 1: figure out what name we had: how? Then, if we have off-site secondaries that have copies of the private zone, we can poll for the highest serial number and copy the contents of that zone. If we only have secondaries for the public part of the zone, we can recover that; should we?

2.9. Well-known names

Homenets serve a zone under the special-use TLD [TBD2], that answers queries for local configuration information and can be used to advertise services provided by the homenet (as opposed to services present on the homenet). This provides a standard means for querying the homenet that can be assumed by management functions and homenet clients. A registry of well-known names for this zone is defined in IANA considerations (Section 8). Names and RRs in this zone are only ever provided by the homenet--this is not a general purpose service discovery zone.

All resolvers on the homenet will answer questions about names in this zone; entries in the zone are guaranteed not to be globally unique. Hosts and services that use the special names under this TLD are assumed to be aware that it is a special TLD. If such hosts cache DNS entries, DNS entries under this TLD are discarded whenever the host detects a network reattachment.

The uuid.[TBD2] name contains a TXT RR that contains the UUID of the homenet. Each homenet generates its own distinct UUID; homenet routers on any particular homenet all use the same UUID, which is agreed upon using HNCP. If the homenet has not yet generated a UUID, queries against this name will return NXDOMAIN.

The global-name.[TBD2] name contains a PTR record that contains the global name of the homenet. If the homenet does not have a global name, queries against this name will return NXDOMAIN.

The global-name-register.[TBD2] name contains one or more A and/or AAAA records referencing hosts that provide a RESTful API over HTTP that can be used to register the global name of the homenet, once that name has been configured.

3. Name Resolution

3.1. Configuring Resolvers

Hosts on the homenet receive a set of resolver IP addresses using either DHCP or RA. IPv4-only hosts will receive IPv4 addresses of resolvers, if available, over DHCP. IPv6-only hosts will receive resolver IPv6 addresses using either stateful (if available) or stateless DHCPv6, or through the domain name option in router advertisements. All homenet routers provide resolver information using both stateless DHCPv6 and RA; support for stateful DHCPv6 and DHCPv4 is optional (right?).

3.2. Configuring Service Discovery

DNS-SD uses several default domains for advertising local zones that are available for service discovery. These include the ".local" domain, which is searched using mDNS, and also the IPv4 and IPv6 reverse zone corresponding to the prefixes in use on the local network. For the homenet, queries against the ".local" zone are supported, as well as queries against every IPv4 address prefix advertised on a homenet link, and every IPv6 address prefix advertised on a homenet link, including prefixes derived from the homenet's ULA(s). In addition, the [TBD2] domain can be used, and is preferred. Whenever the "<domain>" sequence appears in this section, it references each of the domains mentioned in this paragraph.

Homenets advertise the availability of several browsing zones in the "b._dns_sd.<domain>" subdomain. The zones advertised are the "well known" zone (TBD2) and the zone containing the local namespace. If the global name is available, only that name is advertised for the local namespace; otherwise [TBD1] is advertised. Similarly, if the global name is available, it is advertised as the default browsing and service registration domain under "db._dns_sd.<domain>", "r._dns_sd.<domain>", "dr._dns_sd.<domain>" and "lb._dns_sd.<domain>"; otherwise, the name [TBD1] is advertised as the default.

3.3. Resolution of local namespaces

The local namespace appears in two places, under [TBD1] and, if the homenet has a global name, under the global name. Resolution from inside the homenet yields the contents of the local namespaces; resolution outside of the homenet yields the contents of the public namespaces. If there is a global name for the homenet, RRs containing names in both instances of the local namespace are qualified with the global name; otherwise they are qualified with [TBD1].

3.4. Local and Public Zones

The homenet's global name serves both as a unique identifier for the homenet and as a delegation point in the DNS for the zone containing the homenet's forward namespace. There are two versions of the forward namespace: the public version and the private version. Both of these versions of the namespace appear under the global name delegation, depending on which resolver a host is querying.

The homenet provides two versions of the zone. One is the public version, and one is the local version. The public version is never visible on the homenet (could be an exception for a guest net). The

public version is available outside of the homenet. The local version is visible on the homenet. Whenever the zone is updated, it is signed with the ZSK. Both versions of the zone are signed; the local signed version always has a serial number greater than the public signed version. [we want to not re-sign the public zone if no public names in the private zone changed.]

This dual publication model relies on hosts connected to the homenet using the local resolver and not some external resolver. Hosts that use an external resolver will see the public version of the namespace. From a security UI design perspective, allowing queries from hosts on the homenet to resolvers off the homenet is risky, and should be prevented by default. This is because if the user sees inconsistent behavior on hosts that have external resolvers configured, they may attempt to fix this by making all local names public. If an alternate external resolver is to be used, it should be configured on the homenet, not on the individual host.

One way to make this work is to intercept all DNS queries to non-homenet IP addresses, check to see if they reference the local namespace, and if so resolve them locally, answering as if from the remote cache. If the query does not reference a local namespace, and is listed as "do not forward" in RFC 6761 or elsewhere, it can be sent to the intended cache server for resolution without any special handling for the response. This functionality is not required for homenet routers, but is likely to present a better user experience.

3.5. Legacy support

In principle, devices that support DNSSD should be able to do service discovery using DNS without any special help. Devices that only support mDNS should be able to get a complete list of services from a combination of names published by devices on the same link and by the homenet router that serves that link (what if there's more than one?). In cases where the homenet router has an off-link entry that has the same claimed name as an on-link service, the homenet router does not advertise the off-link service.

3.6. DNSSEC Validation

The [TBD1] zone is not validated. We could define a special rule, such that any particular local zone publishes a unique identifier for that zone and signs itself with a ZSK; a homenet-aware host could to TOFU on the id/ZSK, and could keep a list of id/ZSKs it has seen, and then do DNSSEC validation on names in the local zone that way, but it's a bit rickety and nonstandard, so I don't know if there's enough benefit to justify the cost. Worth thinking about, though--could be the keystone of a homenet security model if done right.

3.7. Support for Multiple Provisioning Domains

Homenets must support the Multiple Provisioning Domain Architecture [6]. In order to support this architecture, each homenet router that provides name resolution must provide one resolver for each provisioning domain (PvD). Each homenet router will advertise one resolver IP address for each PvD. DNS requests to the resolver associated with a particular PvD, e.g. using RA options [8] will be resolved using the external resolver(s) provisioned by the service provider responsible for that PvD.

The homenet is a separate provisioning domain from any of the service providers. The global name of the homenet can be used as a provisioning domain identifier, if one is configured. Homenets should allow the name of the local provisioning domain to be configured; otherwise by default it should be "Home Network xxx", where xxx is the generated portion of the homenet's ULA prefix, represented as a base64 string.

The resolver for the homenet PvD is offered as the primary resolver in RAs and through DHCPv4 and DHCPv6. When queries are made to the homenet-PvD-specific resolver for names that are not local to the homenet, the resolver will use a round-robin technique, alternating between service providers with each step in the round-robin process, and then also between external resolvers at a particular service provider if a service provider provides more than one. The round-robining should be done in such a way that no service provider is preferred, so if service provider A provides one caching resolver (A), and service provider B provides two (B1, B2), the round robin order will be (A, B1, A, B2), not (A, B1, B2).

Every resolver provided by the homenet, regardless of which provisioning domain it is intended to serve, will accept updates for services in the local service namespace.

3.8. Using the Local Namespace While Away From Home

Homenet routers do not answer unauthenticated DNS queries from off the local network. However, some applications may benefit from the ability to resolve names in the local namespace while off-network. Therefore hosts connected to the homenet can register keys in the same way that services are registered, and the homenet will cache such keys. Such keys must be validated by the end user before queries against the local namespace that have been authenticated with that key are permitted. A host that will make remote queries to the local namespace caches the names of all DNS servers on the homenet by querying all-resolver-names.[TBD2]. If the local zone is not signed using DNSSEC, the host also caches each server's SIG(0) key.

Hosts that require name resolution from the local network must have a stub resolver configured to contact the dns server on one or more routers in the homenet when resolving names in the local namespace. To do this, resolvers must know the global name of the local namespace, which they can retain from previous connections to the homenet. The homenet may not have a stable IP address, so such resolvers cannot merely cache the IP address of the homenet routers. Instead, they cache the names of the homenet routers that provide DNS service and use those names to determine the IP addresses of the homenet routers at the time of resolution. Such IP addresses can be safely cached for the duration of the TTL of the A or AAAA record that contained them. The names of the homenet router DNS servers should be randomly generated so that they can't be guessed by off-network attackers. [?]

To make a homenet DNS query, the host signs the request using SIG(0) with the key that they registered to the homenet. The homenet router first checks the question in the query for validity: it must be a subdomain of the global name. The homenet router then checks the name of the signing key against the list of cached, validated keys; if that key is cached and validated, then the homenet router attempts to validate the SIG(0) signature using that key. If the signature is valid, then the homenet router answers the query. If the zone is not signed, or doesn't have a trust anchor in the parent zone, the responding server signs the answer with its own SIG(0) key. The resolver that sent the query validates the response using DNSSEC if possible, and otherwise using the SIG(0) key.

[it can be argued that this isn't necessary for the base spec, and it obviously requires some additional protocol work, so may want to leave it out of the base architecture. It may also make more sense to serve queries using DNS-over-TLS (dprive) rather than SIG(0).]

4. Publishing the Public Namespace

4.1. Acquiring the Global Name

There are two ways to acquire a global name: the end-user can register a domain name using a public domain name registry, or the end-user can be assigned a subdomain of a registered domain by a homenet global name service provider. We will refer to this as the Global Name Registration Provider [GNRP]. In either case, the registration process can either be manual or automatic. Homenet routers support automatic registration regardless of the source of the homenet's global name, using a RESTful API.

The RESTful API provides a method for generating a unique URL which can be used for a limited time by the GNRP to register the global

name once the end user has chosen one and made payment arrangements (if necessary). When the GNRP is ready to convey the global name to the homenet, it uses the specified URL to submit (POST) the name. The URL will be https, but the certificate will not be valid; its purpose is to provide privacy, not authenticity.

In response, the homenet server either rejects the POST, if the URL has expired or is invalid, or else returns a text response containing a single label, '@', representing the local namespace. Under that label, the homenet will include one or more DNSKEY records for zone signing keys, one of which is required, and key signing keys, which are not required.

The returned zone will also include a TLSA record. The record has a Usage field value of 0 (PKI Cert), a Selector value of 1 (just the public key), and a matching type of 0 (exact match). The Certificate Association data field contains a public key generated by the homenet for use in authenticating local web traffic.

The returned zone may include NS records; if it does, the GNRP is expected to use those NS records in the delegation for the global name. Otherwise, it provides the NS records for its own authoritative servers.

The GNRP then sets up a secure delegation using the currently-valid ZSKs included in the zone. The GNRP also signs the public key provided in the TLSA record using a PKI cert owned by the GNRP that can be validated by web browsers, and posts it back to the homenet using the RESTful API URL.

More detail on this process will be provided in a future document. [or really, how much detail should there be here?]

4.2. Hidden Primary/Public Secondaries

The default configuration for a homenet's external name service is that the primary server for the zone is not published in an NS record in the zone's delegation. Instead, the GNRP provides authoritative name service for the zone. Whenever the public zone is updated, the hidden primary sends NOTIFY messages to all the secondaries, using the zone's ZSK (or?) to sign the message.

When any of the GNRP secondary servers receives a notify for the zone, it checks to see that the notify is signed with a valid ZSK for that zone. If so, it contacts the IP address from which the NOTIFY was send and initiates a zone transfer. Using this IP address avoids renumbering issues. Upon finishing the zone transfer, the zone is validated using each ZSK used to sign it. If any validation fails,

the new version of the zone is discarded. If updates have been recevied, but no valid updates received, over a user-settable interval defaulting to a day (or?), the GNRP will communicate to the registered user that there is a problem.

The reverse zone for any prefix delegated by an ISP should be delegated by that ISP to the home gateway to which the delegation was sent. The list of secondaries for that zone is sent to the home gateway using DHCPv6 prefix delegation (or?). The ZSK is announced to the ISP in each DHCP PD message sent by the home gateway. Whenever an update is made to this zone, the home gateway sends a NOTIFY to each of the listed secondaries for the delegation, and updates occur as described above.

4.3. DNSSEC security

All zones published by the homenet are signed. Internal zones cannot have secure delegations, however. Hosts that are aware of homenets can do TOFU authentication of a particular instance of the homenet zones [TBD1] or [TBD2]. To do this, the host queries the uuid.[TBD2] name. The homenet always publishes this name with a single TXT RR containing a UUID, which is expected to be unique and stable. The homenet will also publish a name, rev.[TBD2] which contains a PTR RRset that enumerates the outer delegations of all reverse zones operated by the homenet at the time of the query that are in private address spaces.

The homenet-aware host can then query and cache the ZSKs of the [TBD2] domain on that homenet, using the UUID to identify it. The homenet uses the same ZSK for all zones that it publishes. Homenet-aware hosts can validate any record in the [TBD1] and [TBD2] zones and in reverse zones for private and ULA number spaces using the stashed ZSK for the homenet UUID to which the host is currently connected (may be different on different interfaces). Names in non-private, non-ULA number spaces are validated using secure delegations, not homenet TOFU trust anchors, as are all other zones other than [TBD1] and [TBD2].

4.4. PKI security

PKI security is only possible if the homenet has a global name. The homenet should not use TLS unless it has a certificate that will be successfully validated by web servers; otherwise, the homenet will be training end-users to click through certificate warnings, and will be inoperable to web browsers with correct security user interfaces (which never present such warnings). If the homenet has a global name, it should also have gotten a valid PKI certificate as part of the process of acquiring the global name.

The homenet tracks the expiration date of the TLS certificate. One month before expiration, the homenet will send a renewal request to the GNRP using the URL provided by the GNRP during registration. The GNRP will then provide a new certificate and a new URL, which the homenet will record for the next renewal (the URL is not required to change). Because the key to be signed is published in the public namespace of the homenet, there is no need for a secondary authentication path for the key.

4.5. Renumbering

The homenet may renumber at any time. IP address RRs published in either namespace must never have a TTL that is longer than the valid lifetime for the prefix from which the IP address was allocated. If a particular ISP has deprecated a prefix (its preferred lifetime is zero), IP addresses derived from that prefix are not published in the DNS. If more than one prefix is provided by the same ISP and some have different valid lifetimes, only IP addresses in the prefix or prefixes with the longest valid lifetime are.

4.6. ULA

Question: If the homenet has one or more ULAs, should we only publish the ULAs and not the global addresses in the local namespace? This would prevent renumbering events from having any impact on local communication. Any reason not to do this? Would require some rewording of the local/global namespace text.

5. Management

5.1. End-user management

Need to have well-known name with RESTful API that apps can connect to, so that you can have an app on your phone, laptop or whatever that operates the network. This is a model that seems popular and accepted by end-users; having a well-defined API allows us to avoid a million different undocumented vendor-specific management APIs. Web API would also be nice, but we can't specify that, so better to specify the RESTful API and let vendors decide what sort of frock to put on it.

This API should provide a means for notifying end-users of issues on the home network, using whatever app they have installed. The API must provide a mechanism for registering end-users or devices that are permitted to manage the homenet, and a way to recover if all such devices are lost.

5.2. Central management

Possibly can be done mostly through RESTful API, but might want Netconf/Yang as well. Should be possible to have the local namespace mastered on an external DNS auth server, e.g. in case a bunch of HNRs are actually set up in an org, or in case an ISP wants to provide a service package for users who would rather not have an entirely self-operating network.

6. Privacy Considerations

Private information must not leak out as a result of publishing the public namespace. We believe the current provisions adequately address this concern. (right?)

7. Security Considerations

Need someone with security fu to review the registration model, etc., once we have it.

8. IANA considerations

IANA will add a new registry titled Homenet Management Well-Known Names, which initially contains:

uuid Universally Unique Identifier--TXT record containing, in base64 encoding, a stable, randomly generated identifier for the homenet that is statistically unlikely to be shared by any other homenet.

global-name The homenet's global name, represented as a PTR record to that name.

global-name-register The hostname of the homenet's global name registry service, with A and/or AAAA records.

all-resolver-names A list of all the names of the homenet's resolvers for the homenet PvD, represented as an RRset containing one or more PTR records.

The IANA will allocate two names out of the Special-use TLD names registry:

TBD1 Suggested value: "homenet"

TBD2 Suggested value: "_hnsd"

9. Normative References

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

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

[3] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <http://www.rfc-editor.org/info/rfc2136>.

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

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

[6] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <http://www.rfc-editor.org/info/rfc7556>.

[7] Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-03 (work in progress), February 2016.

[8] Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.

Author's Address

Ted Lemon
Nominum, Inc.
800 Bridge Parkway
Redwood City, California 94065
United States of America

Phone: +1 650 381 6000
Email: ted.lemon@nominum.com