Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

T. Lemon
Nominum, Inc.

July 8, 2016

Homenet Naming and Service Discovery Architecture
draft-lemon-homenet-naming-architecture-01

Abstract

This document recommends a naming and service discovery resolution architecture for homenets. This architecture covers local and global publication of names, discusses security and privacy implications, and addresses those implications. The architecture also covers name resolution and service discovery for hosts on the homenet, and for hosts that roam off of the homenet and still need access to homenet services.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

- Provisioning of a domain name under which names can be published and services advertised

- Associating names that are subdomains of that name with hosts.

- Advertising services available on the local network by publishing resource records on those names.

- 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 and resource records 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.

3. 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.

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

5. 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.

6. 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.

7. Devices that provide services must be able to publish those services on the homenet, and those services must be available from any part of the homenet, not just the link to which the device is attached.

8. Homenet explicitly supports multihoming--connecting to more than one Internet Service Provider--and therefore support for multiple provisioning domains [9] 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.

9. 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.

10. To the extent possible, the homenet should support DNSSEC. If the homenet local domain is not unique, there should still be a mechanism that homenet-aware devices can use to bootstrap trust for a particular homenet.

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 identifier.

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 [7] can provide naming and service discovery [8], 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 [4], and a mechanism for discovering services [8]. 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 [10] proposes a mechanism for extending multicast DNS beyond a single multicast domain.. 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. [xxx is this really true at the UI?]

2. Terminology

This document uses the following terms and abbreviations:

HNR Homenet Router

ISP Internet Service Provider

GNRP Global Name Registration Provider

3. 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 infrastructure. This isn't a clean division of responsibility, however. It's possible for the device to maintain change control over its own name, while still performing name resolution for that name in the network infrastructure.

If devices maintain change control on their own names, conflicts can arise. Two devices might present the same name, either because their default names or the same, or as a result of accidental. Devices can be attached to more than one link, in which case we want the same name to identify them on both networks. Although homenets are self-configuring, user customization is permitted and useful, and while some devices may provide a user interface for setting their name, it may be worthwhile to provide a user interface and underlying support for allowing the user to specify a device's name in the homenet infrastructure.

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

3.1. 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 top-level name [TBD1] as their "global name" until they are configured with a global name.

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 subdomain of a domain owned by one of the user's ISP, or managed by some DNS 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.

Homenets must support automatic configuration of the homenet global name in a secure manner, as well as manual configuration of the name. The solution must allow a user with a smartphone application or a user with a web browser to successfully configure the homenet's global name without manual data entry. The security implications of this process must be identified and, to the extent possible, addressed.

3.2. Local namespaces

Every homenet has two or more non-hierarchical local namespaces, one for names of hosts--the host namespace--and one or more for IP addresses--the address namespaces. A namespace is a database table mapping each of a set keys to its value. "Local" in this context means "visible to users of the homenet," as opposed to "public," meaning visible to anyone.

For the host namespace, the key is the set of labels in a name, excluding whatever labels represent the domain name of the namespace. So for example if the homenet's global name is "dog-pixel.example.com" and the name being looked up is "alice.dog-pixel.example.com", the key will be "alice".

The local namespace may be available both in the global DNS namespace and under the [TBD1] special-use name. The set of keys is the same in both cases--in the above example, the name could be either 'alice.dog-pixel.example.com' or 'alice.[TBD1]'. Whichever one of the two representations is used, the key is simply 'alice'.

For each address namespace, the key is the locally-significant portion of the IP address. For example, if the local prefix assigned by an ISP is 2001:DB8:bee7::/48, the name of that address namespace will be '7.e.e.b.8.b.d.0.1.0.0.2.ip6.arpa'. An IP address of 2001:db8:bee7::1 would therefore yield a key of '1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0'

Every prefix in use on the homenet has an address namespace, whether its subdomain is delegated in the DNS or not. This includes any public or private IPv4 prefixes in use [3] as well as any ULA prefixes in use [5], which can't be delegated [6]. When the valid lifetime for a prefix that had been in use on the homenet ends, the address namespace for that prefix is discarded. Namespaces for prefixes that are manually configured, like IPv4 public prefixes and IPv4 private prefixes, persist as long as the prefix is configured. Since ULA prefixes have lifetimes, the lifetime rule applies to their address namespaces.

In all namespaces, the value that the key addresses is a sub-table containing one or more RRsets, each of which is identified by its RRtype. In the terminology of the DNS protocol, each of these namespaces is analogous to a DNS zone (but bear in mind that from the perspective of DNS queries, the namespace for names may appear to hosts connected to the homenet as two different zones containing identical data.

However, in addition to DNS zone data, each RRset also has two metadata flags: the public flag and the critical flag. The public flag indicates whether the data in this RRset should be publicly visible. The critical flag indicates whether the service should be advertised even on high-cost internet links.

Each RR that contains a name (e.g, a CNAME or SRV record) either contains a local name or a name in the public DNS. Local names can be subdomains of the homenet's global name, yet not be public, if no RRsets in the namespace for names is marked public. Local names can also be subdomains of [TBD1]. Names in the public DNS that are not subdomains of the homenet's global name can only be added by explicit action in one of the management interfaces described in Section 6.

Each 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.

3.3. Public namespaces

Every homenet has one or more public namespaces. These are subsets of the local namespaces with the following modifications:

1. Names with no RRsets whose public bits are set 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. Private IPv4 addresses, however, are never published.

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

5. If an RRset contains names, names that are subdomains of either the homenet's global name or [TBD1] are checked in the local host namespace to see if they are marked public. If not, they are omitted.

Because the public namespaces are subsets of the local namespaces, replication is not necessary: each homenet router automatically produces public namespaces by deriving them from the local namespaces using the above rules. Answers to queries in the public namespaces can be generated on demand. However, it may be preferable to maintain these namespaces as if they were DNS zones. This makes it possible to use DNS zone transfers to offload the contents of public zones to a secondary service provider, eliminating the need to handle arbitrary numbers of queries from off of the homenet.

A mechanism will be present that allows devices that have been configured to publicly advertise services to indicate to the homenet that the public bit and/or the backup bit will be set in RRsets that they publish.

3.4. Maintaining Namespaces

Homenets support three methods for maintaining local namespaces. These rely on Multicast DNS, DNS updates, and any of the management mechanisms mentioned in Section 6.

3.4.1. Multicast DNS

HNRs cooperate to maintain a DNS mirror of the set of names published by mDNS. This works similarly to the Multicast DNS Hybrid Proxy [10]. However, the DNSSD hybrid proxy exposes the topology of the network in which it operates to the user.

In order to avoid this, the homenet solution maintains a host namespace for each non-edge link in the homenet. Queries for names in the host namespace are looked up in the per-link host namespaces as well (and trigger mDNS queries as in the hybrid solution). When a cross-link name conflict is present for a name, the name is presented with a short modifier identifying the link.

For example, if two devices on two separate links both advertise the name ’janus’ using mDNS, and the name ’janus’ is not present in the host namespace, the two hosts’ names are modified to, for example, ’janus-1’ and ’janus-2’. If both devices present the human readable name ’Janus’, then that name is presented as ’Janus (1)’ and ’Janus (2)’. If the name ’janus’ appears in the host namespace, then that name is presented just as ’janus’.

If a mDNS service advertises a name that appears in the host namespace, the HNR that hears the advertisement will defend the name, forcing the mDNS service to choose a different name.

This solution shares a problem that mdns hybrid has: user interfaces on hosts that present mDNS names in their mDNS format (e.g., ’janus.local’) will not have a DNS entry for ’janus.local’. Connections to such hosts using the name presented in the UI will work when both hosts are attached to the same link, but not otherwise.

It is preferable that devices that are homenet-aware publish their names using DNS updates rather than using mDNS. mDNS is not supported as a query mechanism on homenets, other than in the sense that homeneds do not filter mDNS traffic on the local link. Service discovery is instead done using DNS service discovery [8]. This mechanism is supported on all modern devices that do service discovery, so there is no need to rely on mDNS.

3.4.2. DNS Update

DNS updates to the resolver on the local link are supported for adding names to local zones. When an update is received, if the name being updated does not exist, or if the update contains the same information as is present in the existing record, then the update is accepted. If a conflicting entry exists, the update is rejected.

This update procedure is available to hosts that implement DNS update for DNS service discovery, but are not homenet-aware. Hosts cannot delete records they have added, nor modify them; such records can only time out. Updates to server list records require that the host referenced by the update exist, and that the update come from that host. Such updates are additive, and are removed automatically when they become stale.

Hosts that are homenet-aware generate a KEY record containing a public key for which they retain the private key. They then publish their name in the host namespace, with whatever data they intend to publish on the name, and include the KEY record they have generated. The update is signed using SIG(0) on the provided key. If a record already exists, and does not contain the same KEY record, the update is refused. Otherwise it is accepted.

Homenet-aware hosts can then update their entries in the address table and in service tables by using their KEY record with SIG(0). Entries can be added _and_ deleted. However, only modifications to RRs that reference the name in the host namespace are allowed; all other RRs must be left as they are.

3.5. 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, which will happen when the device is power cycled, and also may happen when it sees a link state transition. The hybrid mDNS implementation will also discover devices automatically when service queries are made.

Devices that maintain their state using DNS update, but that are not homenet-aware, may or may not update their information when they see a link state transition. Homenet-aware devices will update whenever they see a link-state transition, and also update periodically. When the Homenet configuration has been lost, HNRs advertise a special ND option that indicates that naming and service discovery on the homenet is in a recovery state. Homenet-aware devices will be sensitive to this ND option, and will update when it is seen.

Homenets will present an standard management API, reachable through any homenet router, that allows a device that has stored the DNSSEC ZSK and KSK to re-upload it when it has been lost. This is safest solution for the end user: the keys can be stored on some device they control, under password protection.

ZSKs and KSKs can also be saved by the ISP or GNRP and re-installed using one of the management APIs. This solution is not preferable, since it means that the end user’s security is reliant on the security of the GNRP or ISP’s infrastructure.

If the ZSK and KSK are lost, they can be regenerated. This requires that the homenet’s global name change: there is no secure way to re-key in this situation. Once the homenet has been renamed and re-keyed, all devices that use the homenet will simply see it as a different homenet.

3.6. Well-known names

Homenets serve a zone under the special-use top-level name [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 9). 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: different homenets are guaranteed to give independent and usually different answers to queries against this zone. 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 link state transition.

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 (typically HNRs) 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.

The all-resolver-names.[TBD2] name contains an NS RRset listing a global name for each HNR. It will return NXDOMAIN if the homenet has no global name. These names are generated automatically by each HNR when joining the homenet, or when a homenet to which the HNR is connected establishes a global name.

4. Name Resolution

4.1. 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, however if either service is offered, resolver addresses will be provided using that mechanism as well. Resolver IP addresses will always be IP addresses on the local link: every HNR is required to provide name resolution service. This is necessary to allow DNS update using presence on-link as a mechanism for rejecting off-network attacks.

4.2. 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, no support for queries against the ".local" zone is provided by HNRs: a ".local" query will be satisfied or not by services present on the local link. This should not be an issue: all known implementations of DNSSD will do unicast queries using the DNS protocol.

Service discovery is configured using the technique described in Section 11 of DNS-Based Service Discovery [8]. HNRs will answer domain enumeration ueries 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). 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.

4.3. 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].

4.4. Service Discovery Resolution

Because homenets provide service discovery over DNS, rather than over mDNS, support for DNS push notifications [11]. When a query arrives for a local namespace, and no data exists in that namespace to answer the query, that query is retransmitted as an mDNS query. Data that exists to answer the query in mdns cached namespaces does not prevent an mDNS query being issued.

If there is data available to answer the query in the host namespace or any of the dnssd cached namespaces, that data is aggregated and returned immediately. If the host that sent the query requested push notification, then any mDNS responses that come in subsequent to the initial answer are sent as soon as they are received, and also added to the cache. This means that if a name has been published directly using DNS, no mDNS query for that name is ever generated.

4.5. 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.

4.6. DNSSEC Validation

All namespaces are signed using the same ZSK. The ZSK is signed by a KSK, which is ideally kept offline. Validation for the global name is done using the normal DNSSEC trust hierarchy. Validation for the [TBD1] and [TBD2] zones can be done by fetching the global name from the [TBD2] zone, fetching and validating the ZSK using DNSSEC, and then using that as a trust anchor.

Only homenet-aware hosts will be able to validate names in the [TBD1] and [TBD2] zones. The homenet-aware host validates non-global zones by determining which homenet it is connected to querying the uuid.[TBD2] and global-name.[TBD2] names. If there is an answer for the global-name.[TBD2] query, validation can proceed using the trust anchor published in the zone that delegates the global name. If only the uuid is present, then the homenet-aware host can use trust-on-first-use to validate that an answer came from the homenet that presented that UUID. This provides only a limited degree of trustworthiness.

4.7. Support for Multiple Provisioning Domains

Homenets must support the Multiple Provisioning Domain Architecture [9]. 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 [12] 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 from hosts on the local link.

4.8. 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 host namespace using DNS Update. Such keys must be validated by the end user before queries against the local namespace can be authenticated using that key. 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].

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 host or address namespaces. 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 NS record listing the HNRs 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 doesn’t have a trust anchor in the parent zone, the responding server signs the answer with its own ZSK. The resolver that sent the query validates the response using DNSSEC if possible, and otherwise using the ZSK directly.

5. Publishing the Public Namespace

5.1. 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.

5.2. 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 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. 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. Once the delegation is established, the ISP will not accept a different ZSK unless the prefix and its delegated zone are reassigned.

5.3. PKI security

All communication with the homenet using HTTP is encrypted using opportunistic security. If the homenet is configured with PKI, then the PKI certificate is used. Homenets should automatically acquire a PKI certificate when a global name is established. This certificate should be published in a TLSA record in the host namespace on any hostnames on which HTTP service is offered by HNRs.

5.4. Renumbering

The homenet may renumber at any time. IP address RRs published in any 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 any namespace. 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 published.

5.5. ULA

Homenets have at least one ULA prefix. If a homenet has two ULA prefixes, and one is deprecated, addresses in the second ULA prefix are not published. The default source address selection algorithm ensures that if a service is available on a ULA, that ULA will be used rather than the global address. Therefore, no special effort is made in the DNS to offer only ULAs in response to local queries.

6. Management

6.1. End-user management

Homenets provide two management mechanisms for end users: an HTTP-based user interface and an HTTP-based RESTful API [tbw].

Homenets also provide a notification for end users. By default, when an event occurs that requires user attention, the homenet will attract the user’s attention by triggering captive portal detection on user devices. Users can also configure specific devices to received management alerts using the RESTful management API; in this case, no captive portal notification is performed.

6.2. 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.

7. Privacy Considerations

Private information must not leak out as a result of publishing the public namespace. The ’public’ flag on RRsets in homenet-managed namespaces prevents leakage of information that has not been explicitly marked for publication.

The privacy of host information on the local net is left to hosts. Various mechanisms are available to hosts to ensure that tracking does not occur if it is not desired. However, devices that need to have special permission to manage the homenet will inevitably reveal something about themselves when doing so. It may be possible to use something like HTTP token binding[13] to mitigate this risk.

8. Security Considerations

There are some clear issues with the security model described in this document, which will be documented in a future version of this section. A full analysis of the avenues of attack for the security model presented here have not yet been done, and must be done before the document is published.

9. 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 Domain Names registry:

TBD1 Suggested value: "homenet"

TBD2 Suggested value: "_hnsd"

10. 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] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[4] 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>.

[5] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <http://www.rfc-editor.org/info/rfc4193>.

[6] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, DOI 10.17487/RFC6303, July 2011, <http://www.rfc-editor.org/info/rfc6303>.

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

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

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

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

[11] Pusateri, T. and S. Cheshire, "DNS Push Notifications", draft-ietf-dnssd-push-07 (work in progress), April 2016.

[12] 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.

[13] Popov, A., Nystrom, M., Balfanz, D., Langley, A., and J. Hodges, "Token Binding over HTTP", draft-ietf-tokbind-https-05 (work in progress), July 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