Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 24, 2011

T. Savolainen
Nokia
J. Kato
NTT
T. Lemon
Nominum, Inc.
June 22, 2011

Improved DNS Server Selection for Multi-Homed Nodes
draft-ietf-mif-dns-server-selection-03

Abstract

A multi-homed node can be connected to multiple networks that may utilize different DNS namespaces. The node commonly receives DNS server configuration information from all connected networks. Some of the DNS servers may have information about namespaces other servers do not have. When the multi-homed node needs to utilize DNS, it has to choose which of the servers to contact to. This document describes a policy based method for helping on selection of DNS server, for both forward and reverse DNS lookup procedures, with help of DNS suffix and IPv6 prefix information received via DHCPv6 or DHCPv4.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

A multi-homed node faces several problems a single-homed node does not encounter, as is described in [I-D.ietf-mif-problem-statement]. This document studies in detail the problems local namespaces may cause for multi-homed nodes and provides a solution for IPv6 domain. The node may be implemented as a host or as a router.

When multiple namespaces are visible for a node, some DNS servers have information other servers do not have. Because of that, a multi-homed node cannot assume every DNS server is able to properly answer for any query, but instead the node must be able to ask right server for the information it needs.

An example of an application that benefits from multi-homing is a web browser that commonly accesses many different destinations and needs to be able to dynamically communicate over different network interfaces.

In deployments where multiple namespaces are present, selection of correct route and destination and source addresses for the actual IP connection is crucial as well, as the resolved destination’s IP addresses may be only usable on the network interface over which the name was resolved on. Hence solution described in this document is assumed to be commonly used in combination with tools for delivering additional routing and source and destination address selection policies.

The Appendix A describes best current practices possible with tools preceding this document and on networks not supporting the solution described in this document. As it is possible to solve the problem with less efficient and less explicit manners, the new solution may be considered as an optimization. However, in some environments this solution is considered essential.

1.1. Requirements Language

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

2. Problem description for local namespaces with multi-homed nodes

This chapter describes two host multi-homing related local namespace scenarios for which the procedure described in chapter 3 provides a solution for. Essentially the same challenges may be faced by Consumer Premises Equipment as is described in

[I-D.ietf-v6ops-multihoming-without-nat66]. This chapter additionally describes a related problem for which this document provides only partial solution.

2.1. Fully qualified domain names with limited scopes

A multi-homed node may be connected to one or more networks that are using local namespaces. As an example, the node may have simultaneously open a wireless LAN (WLAN) connection to the public Internet, cellular connection to an operator network, and a virtual private network (VPN) connection to a corporate network. When an application initiates a connection establishment to an FQDN, the host needs to be able to choose the right DNS server for making a successful DNS query. This is illustrated in the figure 1. An FQDN for a public name can be usually resolved with any DNS server, but for an FQDN of corporation’s or operator’s service’s local name the node needs to be able to correctly select the right DNS server for the DNS resolution, i.e. do also network interface selection already before destination’s IP address is known.

```
                            +---------------+
                            | DNS server w/ |    |   Corporate
   +------+                 | public +      |----|   Intranet
   |      |                 | corporation’s |    |
   |      |===== VPN =======| local names   |    |
   |      |                 +---------------+  +----+
   | MIF  |                                    | FW |
   | node |                                    +----+
   |      |                 +---------------+    |
   |      |----- WLAN ------| DNS server w/ |----|   Public
   |      |                 | public names  |    |   Internet
   |      |                 +---------------+  +----+
   |      |                                    | FW |
   |      |                 +---------------+  +----+
   |      |---- cellular ---| DNS server w/ |    |
   +------+                 | public +      |    |   Operator
                            | operator’s    |----|   Intranet
                            | local names   |    |
                            +---------------+
```

Local DNS namespaces illustrated

Figure 1

2.2. Network interface specific IP addresses

In the second problem an FQDN is valid and resolvable via different network interfaces, but to different and not necessarily globally reachable IP addresses, as is illustrated in the figure 2. Node's routing and source and destination address selection mechanism must ensure the destination's IP address is only used in combination with source IP addresses of the network interface the name was resolved on.

```
                            +--------------------|      |
   +------+   IPv6          | DNS server A       |------| IPv6
   |      |-- interface 1 --| saying Peer is     |      |
   |      |                 | at: 2001:0db8:0::1 |      |
   | MIF  |                 +--------------------+   +------+
   | node |                                          | Peer |
   |      |                 +--------------------+   +------+
   |      |   IPv6          | DNS server B       |      |
   |      |-- interface 2 --| saying Peer is     |      |
   +------+                 | at: 2001:0db8:1::1 |------| IPv6
                            +--------------------+      |
```

Local DNS namespaces and different IP addresses for an FQDN on interfaces 1 and 2.

Figure 2

Similar situation can happen with IPv6 protocol translation and AAAA record synthesis [RFC6147]. A synthesised AAAA record is guaranteed to be valid only on a network interface it was synthesized on. Figure 3 illustrates a scenario where the peer's IPv4 address is synthesized into different IPv6 addresses by DNS servers A and B.

```
                            +-------------------|    +-------+
   +------+   IPv6          | DNS server A      |----| NAT64 |
   |      |-- interface 1 --| saying Peer is    |    +-------+
   |      |                 | at: A_Pref96:IPv4 |       |
   | MIF  |                 +-------------------+       |   +------+
   | node |                                        IPv4 +---| Peer |
   |      |                 +-------------------+       |   +------+
   |      |   IPv6          | DNS server B      |       |
   |      |-- interface 2 --| saying Peer is    |    +-------+
   +------+                 | at: B_Pref96:IPv4 |----| NAT64 |
                            +-------------------+    +-------+
```

AAAA synthesis results in interface specific IPv6 addresses.

Figure 3

A thing worth noting is that interface specific IP addresses can cause problems also for a single-homed host, if the host retains its DNS cache during movement from one network interface to another. After the interface change a host could have both positive and negative DNS cache entries no longer valid on the new network interface. Because of this the cached DNS information should be considered network interface local instead of node global.

2.3. A problem not fully solved by the described solution

A more complex scenario is an FQDN, which in addition to possibly resolving into network interface specific IP addresses, identifies on different network interfaces completely different peer entities with potentially different set of service offerings. In even more complex scenario, an FQDN identifies unique peer entity, but one that provides different services on its different network interfaces. The solution described in this document is not able to tackle these higher layer issues. In fact, these problems may be solvable only by manual user intervention.

However, when DNSSEC is used, the DNSSEC validation procedure may provide assistance for selecting correct responses for some, but not all, use cases. A node may prefer to use the DNS answer that validates with the preferred trust anchor.

3. Deployment scenarios

This document has been written with three particular deployment scenarios in mind. First being a Consumer Premises Equipment (CPE) with two or more uplink VLAN connections. Second scenario involves a cellular device with two uplink Internet connections: WLAN and cellular. Third scenario is for VPNs, where use of local DNS server may be preferred for latency reasons, but corporate DNS server must be used to resolve private names used by the corporation.

3.1. CPE deployment scenario

A home gateway may have two uplink connections leading to different networks, as is described in [I-D.ietf-v6ops-multihoming-without-nat66]. In this scenario only first uplink connections lead to Internet, while second uplink connection leads to a private network utilizing private namespace.

It is desirable that the CPE does not have to send DNS queries over both uplink connections, but instead CPE should send default queries to the DNS server of the interface leading to the Internet, and queries related to private namespace to the DNS server of the private network.

In this scenario the legacy hosts can be supported by deploying DNS proxy on the CPE and configuring hosts in the LAN to talk to the DNS proxy. However, updated hosts would be able to talk directly to the correct DNS servers of each uplink ISP’s DNS server. It is deployment decision whether the updated hosts would be pointed to DNS proxy or to actual DNS servers.

Depending on actual deployments, all VLAN connections may be considered secure.

3.2. Cellular network scenario

A cellular device may have both WLAN and cellular network interfaces up. In such a case it is often desirable to use WLAN by default, except for those connections cellular network operator wants to go over cellular interface. The cellular network may utilize private names and hence the cellular device needs to ask for those through the cellular interface.

In this scenario cellular interface can be considered secure and WLAN often insecure.

3.3. VPN scenario

Depending on a deployment, there may be need to use VPN only for traffic destined to a corporate network. The corporation may be using private namespace, and hence related DNS queries should be send over VPN to the corporate DNS server, while by default a DNS server of a local access network may be used.

In this scenario VPN interface can be considered secure and local access network insecure.

3.4. Dual-stack accesses

A node may be connected to one or more dual-stack capable access networks. In such a case both or either of DHCPv4 and DHCPv6 can be used to learn DNS server selection information.

4. Improved DNS server selection

This chapter describes a procedure that a (stub / proxy) resolver may utilize for improved DNS server selection in face of multiple namespaces and multiple simultaneously active network interfaces.

4.1. Procedure for prioritizing DNS servers and handling responses

A resolver SHALL build a priority list of DNS servers it will contact to depending on the query. To build the list in an optimal way, a node SHOULD ask with DHCP which DNS servers of each network interface are most likely able to successfully serve forward lookup requests matching to specific DNS suffixes or reverse (PTR record) lookup requests matching to specific IPv6 prefixes. For security reasons the DNS server selection information MUST be used only when it is safe to do so, see section 4.3 for details.

The node SHOULD create a host specific route for the DNS server addresses learned via DHCP. The route must point to the interface DNS server address was learned on. This is required to ensure DNS queries are sent out via the right interface.

A resolver lacking more explicit information shall assume that all information is available from any DNS server of any network interface. The DNS servers learnt by other DNS server address configuration methods MUST be handled as medium priority default servers.

When a DNS query needs to be made, the resolver SHOULD give highest precedence to the DNS servers explicitly known to serve matching suffixes or prefixes. However, the resolver SHOULD take into account different trust levels of pieces of DNS server selection information the resolver may have received from node’s network interfaces. The resolver SHOULD prefer DNS servers of trusted interfaces. The DNS servers of trusted interfaces may be of highest priority only if trusted interfaces specifically configure DNS servers to be of low priority. The non-exhaustive list on figure 4 illustrates how the different trust levels of received DNS server selection information

SHOULD influence the DNS server selection logic.

A resolver SHOULD prioritize between equally trusted DNS servers with help of the DHCP option preference field. The resolver SHOULD NOT prioritize less trusted DNS servers higher than trusted, even in the case of less trusted server would apprently have additional information. In the case of all other things being equal the resolver shall make the prioritization decision based on its internal preferences.

Information from from more trusted interface A	Information from less trusted interface B	Resulting DNS server priority selection
1. Medium priority default	Medium priority default	Default: A, then B
2. Medium priority default	High priority default High priority specific	Default: A, then B Specific: A, then B
3. Low priority default	Medium priority default	Default: B, then A
4. Low priority default High priority specific	Medium priority default	Default: B, then A Specific: A, then B

Figure 4: DNS server selection in case of different trust levels

The resolver SHOULD avoid sending queries to different interfaces in parallel as that may waste resources, sometimes significantly, and would also unnecessary reveal information about ongoing communications. Independently of whether DNS queries are sent in series or parallel, replies for DNS queries MUST be waited until acceptable positive reply is received, all replies are received, or time out occurs.

Because DNSSEC provides cryptographic assurance of the integrity of DNS data, data that can be validated under DNSSEC is necessarily to be preferred over data that cannot be. It follows that, if validation is not performed by the host making the decision about whether to trust the DNS data from a given interface, it cannot make a decision to prefer data from any interface with any great assurance: any response could be forged, and there is no way to detect it without DNSSEC. Specifically, the validating security aware host MUST NOT proceed with a reply that cannot be validated with DNSSEC if DNS queries sent to other servers are still pending.

In the case of a trusted DNS server replying negatively to a question having matching suffix, it will be for implementation to decide whether to consider that as a final response, or whether to ask also from other DNS servers. The implementation decision may be based, for example, on deployment or trust models.

(DISCUSS: What about those DNS servers that instead of negative answer always return positive reply with an IP address of some captive portal?)

4.2. DNS server selection DHCPv6 option

DHCPv6 option described below can be used to inform resolvers which DNS server should be contacted when initiating forward or reverse DNS lookup procedures.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  OPTION_DNS_SERVER_SELECT     |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            DNS-recursive-name-server (IPv6 address)           |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|prf| Reserved  |                                               |
+-+-+-+-+-+-+-+-+          DNS suffixes and prefixes            |
|                          (variable length)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code: OPTION_DNS_SERVER_SELECT (TBD)

option-len: Lenght of the option in octets

DNS-recursive-name-server: An IPv6 address of a DNS server

prf: DNS server preference, for selecting between equally trusted DNS servers:

    01 High
    00 Medium
    11 Low
    10 Reserved

Reserved: Flags reserved for the future. MUST be set to zero.

DNS suffixes and prefixes: The list of DNS suffixes for forward DNS lookup and prefixes for reverse DNS lookup the DNS server has special knowledge about. Field MUST be encoded as specified in section "Representation and use of domain names" of [RFC3315]. Special suffix of "." is used to indicate capability to resolve global names and act as a default name server. Lack of "." suffix on the list indicates DNS server only has information related to listed suffixes and prefixes. Prefixes for reverse mapping are encoded as defined for ip6.arpa [RFC3152].

DHCPv6 option for explicit DNS suffix configuration

Figure 5

A node SHOULD include an OPTION_ORO option in a DHCPv6 request with the OPTION_DNS_SERVER_SELECT option code to inform the DHCPv6 server about the support for the improved DNS server selection logic. DHCPv6 server receiving this information MAY then choose to provision DNS server addresses only with the OPTION_DNS_SERVER_SELECT.

The OPTION_DNS_SERVER_SELECT contains one or more DNS suffixes the related DNS server has particular knowledge of. The option can occur multiple times in a single DHCPv6 message, if multiple DNS servers are to be configured.

IPv6 prefixes should cover all the DNS suffixes configured in this option. Prefixes should be as long as possible to avoid potential collision with information received on other option instances or with options received from DHCPv6 servers of other network interfaces. Overlapping IPv6 prefixes are interpreted so that the resolver can use any of the DNS servers for queries mathing the prefixes.

If the OPTION_DNS_SERVER_SELECT contains a DNS server address already learned from other DHCPv6 servers and possibly through other network interfaces, the node MAY append new prefixes and suffixes to the information received earlier. The node MUST NOT remove previously obtained information. However, the node SHOULD NOT extent lifetime of earlier information either. In the case conflicting DNS server address and related information is learned from less trusted interface, the node MAY choose to ignore the option.

As the DNS options of [RFC3646], the OPTION_DNS_SERVER_SELECT option MUST NOT appear in any other than the following DHCPv6 messages: Solicit, Advertise, Request, Renew, Rebind, Information-Request, and Reply.

The information conveyed in OPTION_DNS_SERVER_SELECT is considered valid until changed or refreshed by general events that trigger DHCPv6 action. In the event that it is desired for the client to request a refresh of the information, use of generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242] is envisaged.

4.3. DNS server selection DHCPv4 option

DHCPv4 option described below can be used to inform resolvers which DNS server should be contacted when initiating forward or reverse DNS lookup procedures.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     CODE      |      Len      | Suffix count  | Reserved  |prf|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            DNS-recursive-name-server (IPv4 address)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                          DNS suffixes and prefixes            |
|                          (variable length)                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

option-code:   OPTION_DNS_SERVER_SELECT (TBD)

option-len:    Lenght of the option in octets

Suffix count:  Number of suffixes and prefixes included

DNS-recursive-name-server: An IPv4 address of a DNS server

prf:           DNS server preference, for selecting between
               equally trusted DNS servers:
                   01 High
                   00 Medium
                   11 Low
                   10 Reserved

Reserved:      Flags reserved for the future. MUST be set to zero.

DNS suffixes and prefixes:  The list of DNS suffixes for forward DNS
               lookup and prefixes for reverse DNS lookup the DNS server
               has special knowledge about. Field MUST be encoded as
               specified in section "Representation and use of
               domain names" of [RFC3315].
               Special suffix of "." is used to indicate
               capability to resolve global names and act as a
               default name server. Lack of "."
               suffix on the list indicates DNS server only has
               information related to listed suffixes and prefixes.
               Prefixes for reverse mapping are encoded as
               defined for in-addr.arpa [RFC2317]. Trailing zeros
               shall be added until next octet boundary.
```

DHCPv4 option for explicit DNS suffix configuration

Figure 6

The OPTION_DNS_SERVER_SELECT contains one or more DNS suffixes the related DNS server has particular knowledge of. The option can occur multiple times in a single DHCPv4 message, if multiple DNS servers are to be configured.

If multiple instances of OPTION_DNS_SERVER_SELECT are present, then the data portions of all the options are concatenated together as specified in "Encoding Long DHCP Options in the Dynamic Host Configuration Protocol (DHCPv4)" [RFC3396].

If the OPTION_DNS_SERVER_SELECT contains a DNS server address already learned from other DHCPv4 servers and possibly through other network interfaces, the node MAY append new prefixes and suffixes to the information received earlier. The node MUST NOT remove previously obtained information. However, the node SHOULD NOT extent lifetime of earlier information either. In the case conflicting DNS server address and related information is learned from less trusted interface, the node MAY choose to ignore the option.

4.4. Limitations on use

A node MAY use OPTION_DNS_SERVER_SELECT in any of the following four cases. In other cases the node MUST NOT use OPTION_DNS_SERVER_SELECT unless the node is specifically configured to do so.

1. The server selection option is delivered across a secure, trusted channel.

2. The server selection option is not secured, but the client on a node does DNSSEC validation.

3. The server selection option is not secured, the resolver does DNSSEC validation, and the client communicates with the resolver configured with server selection option over a secure, trusted channel.

4. The DNS server IP address that is being recommended in the server selection option is known and trusted by the client; that is, the server selection option serves not to introduce the client to a new server, but rather to inform it that a server it has already been configured to trust is available to it for resolving certain domains.

4.5. Coexistence with RFC3646

The OPTION_DNS_SERVER_SELECT is designed to coexist with OPTION_DNS_SERVERS defined in [RFC3646]. The DNS servers configured via OPTION_DNS_SERVERS MUST BE considered as default name servers with medium preference. When both options are received from the same network interface and the OPTION_DNS_SERVER_SELECT contains default DNS server address, the resolver MUST make the decision which one to prefer based on preferences. If OPTION_DNS_SERVER_SELECT defines medium preference then DNS server from OPTION_DNS_SERVER_SELECT SHALL be selected. All default servers are assumed to be able to resolve queries for global names.

If both OPTION_DNS_SERVERS and OPTION_DNS_SERVER_SELECT contain the same DNS server(s) IPv6 address(es), only one instance of each DNS servers’ IPv6 addresses shall be added to the DNS server list.

If a node had indicated support for OPTION_DNS_SERVER_SELECT in DHCPv6 request, the DHCPv6 server may choose to omit sending of OPTION_DNS_SERVERS. This enables offloading use case where network administrator wishes to only advertise low priority default DNS servers.

4.6. Interactions with OPTION_DOMAIN_LIST

A node may be configured with DNS search list with OPTION_DOMAIN_LIST. Resolution for the name containing any dots SHOULD first be attempted with DNS servers of all interfaces as described earlier. Only if the resolution fails the node SHOULD append the name with search list suffix(es) and then utilize improved DNS server selection algorith again to decide which DNS server(s) to contact next. A name without any dots SHALL immediately be appended with suffix(es) and improved DNS server selection be utilized on resolution.

4.7. CNAME/DNAME record considerations

If a node receives a reply with a canonical name (CNAME) or delegation name (DNAME) the follow-up queries MUST be sent to the same DNS server irrespectively of the FQDN received. Otherwise referrals may fail.

5. Example of a node behavior

Figure 6 illustrates node behavior when it initializes two network interfaces for parallel usage and learns DNS suffix and prefix information from DHCPv6 servers.

```
    Application    Node      DHCPv6 server   DHCPv6 server
                             on interface 1  on interface 2
        |             |                |
        |         +-----------+        |
   (1)  |         | open      |        |
        |         | interface |        |
        |         +-----------+        |
        |             |                |
   (2)  |             |---option REQ-->|
        |             |<--option RESP--|
        |             |                |
        |         +-----------+        |
   (3)  |         | store     |        |
        |         | suffixes  |        |
        |         +-----------+        |
        |             |                |
        |         +-----------+        |
   (4)  |         | open      |        |
        |         | interface |        |
        |         +-----------+        |
        |             |                |                |
   (5)  |             |---option REQ------------------->|
        |             |<--option RESP-------------------|
        |             |                |                |
        |         +----------+         |                |
   (6)  |         | store    |         |                |
        |         | suffixes |         |                |
        |         +----------+         |                |
        |             |                |                |
```

Illustration of learning DNS suffixes

Figure 7

Flow explanations:

1. A node opens its first network interface

2. The node obtains DNS suffix and IPv6 prefix information for the new interface 1 from DHCPv6 server

3. The node stores the learned DNS suffixes and IPv6 prefixes for later use

4. The node opens its seconds network interface 2

5. The node obtains DNS suffix, say 'example.com', and IPv6 prefix information, say '8.b.d.0.1.0.0.2.ip6.arpa' for the new interface

2 from DHCPv6 server

6. The node stores the learned DNS suffixes and prefixes for later use

Figure 7 below illustrates how a resolver uses the learned suffix information. Prefix information use for reverse lookups is not illustrated, but that would go as the figure 7 example.

```
    Application     Node     DHCPv6 server     DHCPv6 server
                             on interface 1    on interface 2
        |             |                |                |
   (1)  |--Name REQ-->|                |                |
        |             |                |                |
        |      +----------------+      |                |
   (2)  |      | DNS server     |      |                |
        |      | prioritization |      |                |
        |      +----------------+      |                |
        |             |                |                |
   (3)  |             |------------DNS resolution------>|
        |             |<--------------------------------|
        |             |                |                |
   (4)  |<--Name resp-|                |                |
        |             |                |                |
```

Example on choosing interface based on DNS suffix

Figure 8

Flow explanations:

1. An application makes a request for resolving an FQDN, e.g. 'private.example.com'

2. A node creates list of DNS servers to contact to and uses configured DNS server information and stored DNS suffix information on priorization decisions.

3. The node has chosen interface 2, as from DHCPv6 it was learned earlier that the interface 2 has DNS suffix 'example.com'. The node then resolves the requested name using interface 2's DNS server to an IPv6 address

4. The node replies to application with the resolved IPv6 address

6. Scalability considerations

The size limitations of DHCPv6 messages limit the number of suffixes and prefixes that can be carried in a configuration option. Including the suffixes and prefixes in a DHCPv6 option is best suited for deployments where relatively few carefully selected suffixes and prefixes are adequate.

7. Considerations for network administrators

Network administrators deploying private namespaces should assist advanced hosts in the DNS server selection by providing information described in this document for nodes. To ensure nodes' routing and source and destination IP address selection also works correctly, network administrators should also deploy related technologies for that purpose.

The solution described herein is best for selecting a DNS server having knowledge of some namespaces. The solution is not able to make the right decision in a scenario where the same name points to different services on different network interfaces, as described in section 2.3. Network administrators are recommended to avoid overloading of namespaces in such manner.

To mitigate against attacks against local namespaces, administrators utilizing this tool should deploy DNSSEC for their zone.

8. Acknowledgements

The author would like to thank following people for their valuable feedback and improvement ideas: Mark Andrews, Jari Arkko, Marcelo Bagnulo, Stuart Cheshire, Lars Eggert, Tomohiro Fujisaki, Peter Koch, Suresh Krishnan, Edward Lewis, Kurtis Lindqvist, Arifumi Matsumoto, Erik Nordmark, Steve Padgett, Fabien Rapin, Dave Thaler, Margaret Wasserman, Dan Wing, and Dec Wojciech. Ted Lemon and Julien Laganier receive special thanks for their contributions to security considerations.

This document was prepared using xml2rfc template and the related web-tool.

9. IANA Considerations

This memo includes a new DHCPv6 option that requires allocation of a new code point.

10. Security Considerations

It is possible that attackers might try to utilize OPTION_DNS_SERVER_SELECT option to redirect some or all DNS queries sent by a resolver to undesired destinations. The purpose of an attack might be denial-of-service, preparation for man-in-the-middle attack, or something akin.

Attackers might try to lure specific traffic by advertising DNS suffixes and prefixes from very small to very large scope or simply by trying to place attacker's DNS server as the highest priority default server.

The main countermeasure against these attacks is to use this option only when safe to do so, see section 4.3 for deatils. The safest approach is for nodes to implement validating DNSSEC aware resolvers. Trusting on validation done by a DNS server is a possibility only if a host trusts the DNS server and can use a secure channel for DNS messages.

Decision on trust levels of network interfaces depends very much on deployment scenario and types of network interfaces. For example, unmanaged WLAN may be considered less trustworthy than managed cellular or VPN connections.

A node that accepts DNS server selection rules from non-trusted interfaces and implements DNSSEC validation SHOULD send queries also to (all) other known DNS servers in case a non-validatable response is received from the preferred DNS server. This protects against possible redirection attacks.

11. References

11.1. Normative References

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

[RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 2317, March 1998.

[RFC3152] Bush, R., "Delegation of IP6.ARPA", BCP 49, RFC 3152, August 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

11.2. Informative References

[I-D.ietf-mif-problem-statement] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", draft-ietf-mif-problem-statement-15 (work in progress), May 2011.

[I-D.ietf-v6ops-multihoming-without-nat66] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-ietf-v6ops-multihoming-without-nat66-00 (work in progress), December 2010.

[I-D.wing-behave-dns64-config] Wing, D., "IPv6-only and Dual Stack Hosts on the Same Network with DNS64", draft-wing-behave-dns64-config-03 (work in progress), February 2011.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, November 2002.

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC5006] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Option for DNS Configuration", RFC 5006, September 2007.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

Appendix A. Best Current Practice for DNS server selection

On some local namespace deployments explicit policies for DNS server selection are not available. This section describes ways for hosts to mitigate the problem by sending wide-spread queries and by utilizing possibly existing indirect information elements as hints.

A.1. Sending queries out on multiple interfaces in parallel

A possible current practice is to send DNS queries out of multiple interfaces and pick up the best out of the received responses. A host SHOULD implement DNSSEC in order to be able to reject responses that cannot be validated. Selection between legitimate answers is implementation specific, but replies from trusted servers should be preferred.

A downside of this approach is increased consumption of resources. Namely power consumption if an interface, e.g. wireless, has to be brought up just for the DNS query that could have been resolved also via cheaper interface. Also load on DNS servers is increased. However, local caching of results mitigates these problems, and a node might also learn interfaces that seem to be able to provide 'better' responses than other and prefer those - without forgetting fallback required for cases when node is connected to more than one network using local namespaces.

Another downside is revealing to all DNS servers the names a host is connecting to. For example, a DNS server of a public hotspot could learn all the private names host is trying to connect on other interfaces.

A.2. Search list option for DNS forward lookup decisions

A host can learn the special DNS suffixes of attached network interfaces from DHCP search list options; DHCPv4 Domain Search Option number 119 [RFC3397] and DHCPv6 Domain Search List Option number 24 [RFC3646]. The host behavior is very similar as is illustrated in the example at section 3.3. While these DHCP options are not intented to be used in DNS server selection, they may be used by the host as hints for smarter DNS server prioritization purposes in order to increase likelyhood of fast and successful DNS query.

Overloading of existing DNS search list options is not without problems: resolvers would obviously use the DNS suffixes learned from search lists also for name resolution purposes. This may not be a problem in deployments where DNS search list options contain few DNS suffixes like ’example.com, private.example.com’, but can become a problem if many suffixes are configured.

A.3. More specific routes for reverse lookup decision

[RFC4191] defines how more specific routes can be provisioned for hosts. This information is not intented to be used in DNS server selection, but nevertheless a host can use this information as a hint about which interface would be best to try first for reverse lookup procedures. A DNS server configured via the same interface as more specific routes is more likely capable to answer reverse lookup questions correctly than DNS server of an another interface. The likelyhood of success is possibly higher if DNS server address is received in the same RA [RFC5006] as the more specific route information.

A.4. Longest matching prefix for reverse lookup decision

A host may utilize the longest matching prefix approach when deciding which DNS server to contact for reverse lookup purposes. Namely, the host may send a DNS query to a DNS server learned over an interface having longest matching prefix to the address being queried. This approach can help in cases where ULA [RFC4193] addresses are used and when the queried address belongs to a host or server within the same network (for example intranet).

Appendix B. DNSSEC and multiple answers validating with different trust anchors

When validating DNS answers with DNSSEC, a validator might order the list of trust anchors it uses to start validation chains, in terms of the host’s preferences for those trust anchors. A host could use this ability in order to select among alternative DNS results from different interfaces. Suppose that a host has a trust anchor for the public DNS root, and also has a special-purpose trust anchor for example.com. An answer is received on interface i1 for www.example.com, and the validation for that succeeds by using the public trust anchor. Also, an answer is received on interface i2 for www.example.com, and the validation for that succeeds by using the trust anchor for example.com. In this case, the host has evidence for relying on i2 for answers in the example.com zone.

Authors' Addresses

Teemu Savolainen
Nokia
Hermiankatu 12 D
TAMPERE,   FI-33720
FINLAND

Email: teemu.savolainen@nokia.com

Jun-ya Kato
NTT
9-11, Midori-Cho 3-Chome Musashino-Shi
TOKYO,   180-8585
JAPAN

Email: kato@syce.net

Ted Lemon
Nominum, Inc.
2000 Seaport Boulevard
Redwood City,   CA 94063
USA

Phone: +1 650 381 6000
Email: Ted.Lemon@nominum.com