Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: January 15, 2014

C. Dessez
Cisco Systems

July 14, 2013

Connecting Home Networks via the social network GooglePlus
draft-dessez-homenet-googleplus-interconnect-00

Abstract

This document describes an experimental implementation for connecting home networks via a social network. The social network is used to extend the boundary of a single home network to include other home networks. In this way, access to devices or services within a home can be granted among home networks based on their relation to one another within the social network.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
   1.1. Requirements Language 3
   1.2. Terminology and abreviations 3
2. Defining the set of connected homes 4
3. Overall architecture 5
4. Network architecture 6
   4.1. Managing the tunnels 6
   4.2. Configuring the network 7
5. Sharing services within your set of connected homes 8
6. Security Considerations 9
7. Experimental results 10
8. IANA Considerations 10
9. Acknowledgements 10
10. References 11
   10.1. Normative References 11
   10.2. Informative References 12
Author’s Address 12

1. Introduction

The goal of this experiment is to allow an average home user to extend the boundaries of their home network to other home networks the user trusts. Other home networks may be owned by a single user, or "friends" of the user as defined by a social network. This document describes an overall architecture and specific mechanisms chosen for a working implementation based on the social network Google Plus.

In each home, one router is responsible for interacting with the social network. The home network is represented within the social network as a "Page" which the user owns. The router is given credentials to interact with its representative Page, while the user defines the relationship of the Pages to one another. When a bidirectional relationship between two home network Pages is detected, the information necessary to setup a tunnel is shared by posting it to the social network. An encrypted tunnel is then setup between the homes, and a link established.

IP reachability among linked homes is achieved by insertion and propagation of routes into a routing protocol running within the home network. Services are then advertised among homes as defined in [I-D.cheshire-mdnsext-hybrid] and [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]. Finally, by connecting to a UI hosted by the specific router, the user can define policies for the services permitted to be shared within a given circle defined by the social network.

The mechanisms described in the following sections assume a homenet environment as described in [I-D.ietf-homenet-arch] with with a routing protocol such as that defined in ([I-D.acee-ospf-ospfv3-autoconfig]) as well as the mechanism of prefix assignment defined in [I-D.arkko-homenet-prefix-assignment] .

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].

1.2. Terminology and abreviations

In this section we define terminology and abbreviations used throughout the text.

- Homenet: a home network as defined in [I-D.ietf-homenet-arch]

- Gplus: Google Plus. Google’s social network.

- Gplus router: the router that is responsible for the connection to Google Plus, on which the mechanisms described in this document are hosted.

- Circle: represents a group of people for which you can define confidentiality and visibility policies in Google Plus.

- Gplus ID: the unique internal identifier of an entity in Google Plus. It apparently consists in a decimal number on the order of 10^21 for users and pages accounts, and a 64-bit hexadecimal number for circles.

- DNS-SD: DNS-Based Service Discovery [RFC6763].

- ULA: IPv6 Unique Local Addresses [RFC4193].

- CA: Certificate Authority (as defined in TLS [RFC5246]).

- CRT: Certificate as defined in TLS [RFC5246]

- CSR: Certificate Signing Request or Certificate Request (as defined in TLS [RFC5246]).

- CPE: Customer Premises Equipment.

2. Defining the set of connected homes

The central idea of this experiment is for the homenet to be represented within the social network in a way that is intuitive to the user. For this to happen, the homenet must be represented in a way such that:

- the homenet is clearly linked to its owner

- the user can manage the relationships of the homenet with other homenets linked to other users

- the network devices in the homenet can retrieve its social topology and setup communication with its related homenets

If social networks were widely used for connecting homenets today, there may be some specific entity that a user could define that would clearly be identified as a home network. This would be available for setting up connections to, based on the users policy and relationship to other users with homenets as part of their profile. As that is not the case today among popular social networks such as Facebook and Google Plus, we looked into what might be the closest fit and decided to use Google Plus pages. Intended mainly for brands and businesses, they are not very different from user accounts on a social point of view (they organize their contacts and what they see by the system of circles). A user may have several pages, and a page may have several administrators, each of them being able to easily log in as the page while connected to their regular Gplus account.

In this implementation, the home router connects to Gplus to retrieve the topology and communicate with other routers using the Google Pages API. This API uses OAuth 2.0 ([RFC6749]) to allow the user to delegate the management of pages to their Gplus router.

In Gplus, the relationships between people and pages are ruled by the system of circles. One can circle whoever they want in one or more of their circles, without it needing to be accepted by the latter. But in our case, we consider a tunnel must be created only if the relationship is bidirectional, that is only if they have both circled each other in at least one circle. Notice that whereas one cannot know what are the circles of someone else, they know who has circled them, which is enough to know whether a relationship is bidirectional. The Section 5 will explain in details how the visibility policies of DNS-SD services are directly linked to circles.

As stated earlier, the router needs to send messages through Gplus in order to exchange the information necessary to establish and configure the tunnel. This information can be divided into three categories: routing information, cryptographic keys and DNS-SD settings. The first two will be detailed in Section 4 and the third one in Section 5. The routing information and the DNS-SD settings, which we will call Network Setttings, are gathered in a post that is regularly updated and visible to everyone in the page’s circles. The posts conveying cryptographic will be described in Section 4.

3. Overall architecture

Figure 1 represents the global functional architecture of the implementation and shows the interactions between its different parts.

The interaction with Gplus is handled by a module called GplusHandler. It performs regular polling to update the social topoly in the database, and provides the TunnelsManager with functions which can send and retrieve messages or force an update of the database.

The TunnelsManager is responsible for launching and maintaining the tunnels. It also takes care of routing and network settings issues.

A user interface enables the user to modify the service policies stored in the database. Thus, they can be accessed by the FirewallManager and the customized DNS server that filters DNS-SD requests accordingly.

```
                                +-------+
                                | Gplus |
                                +-------+
                                    |
                                    |
                            +--------------+   +----------------+
                            | GplusHandler |---| TunnelsManager |
                            +--------------+   +----------------+
                                    |         /
                                    |        /
         +------------------+       |       /
         | UI to define     |   +----------+    +-----------------+
         | service policies |---| Database |----| FirewallManager |
         +------------------+   +----------+    +-----------------+
                                    |          /
                                    |         /
                             +---------------+
                             |  DNS Server:  |
                             | DNS-SD filter |
                             +---------------+
```

Overall functional architecture.

Figure 1

4. Network architecture

4.1. Managing the tunnels

The tunnelling technology chosen for this experiment is OpenVPN with the cryptography library OpenSSL.

In OpenVPN, one end has to be a server listening to the connections of clients, which in this case are the Gplus routers of the connected homenets. A server might have several clients connected to the same network interface. Notice it can be configured such as the clients connected to the same server cannot send packets to each other. Though there might be better ways to proceed, for this experiment the choice of being server or client is made by comparing the Gplus IDs of the connected pages.

To set a tunnel with proper authentication of the other end, an architecture of OpenSSL certificates must be built. A Certificate Authority (CA) is built and owned by the server which must sign certificates to the clients. The certificates contain the Common Names of they owners, which define the identity of the tunnel endpoints. For this experiment, the Common Name of a router is its Gplus ID. Since each Gplus router may potentially host at the same time a server and multiple clients, it creates a CA and a Certificate Request (CSR). Then it publishes in Gplus a post (here called Security Settings) containing the certificate (CRT) of its CA and its CSR and makes it visible by all its circles. Therefore, when a new relationship appears in the social network, the server retrieves the client’s CSR, signs it with the key of its CA and sends it back with a restricted visibility to the client. As for the client, it retrieves the CRT of the server’s CA and its signed CRT. Notice there is no cryptography key sent on the social network, which is otherwise a secure channel to exchange the CAs and CSRs.

Concerning contact addresses, the Gplus router must have a globally reachable IP address whether IPv4 (for example being the CPE) or preferably IPv6. This/these addresse(s) are advertised in the Network Settings post which is published at boot time and regularly updated, and visible by all the circles of the homenet.

4.2. Configuring the network

In order to enable reachability of the devices of a connected homenet via the tunnel between them, routes must be configured. For reasons explained in Section 6, instead of injecting routes to the globally routable prefixes of the connected homenets, the described design makes the Gplus routers generate and asign ULA prefixes and only those are advertised.

In order to reduce the odds of collision, the ULA prefix is generated by the Gplus router following the following schema:

```
                  |  8 bits      |  40 bits               |
                  +--------------+------------------------+
                  |  FD00::/8    |   Global ID            |
                  +--------------+------------------------+
      Global ID =  f( hash( timestamp + GplusID ) )
```

With:

```
      f          A function that take only the 40 last bits of its
                 argument

      hash       A hashing function (SHA1 for this experiment)

      timestamp  A string containing the current UNIX timestamp

      GplusID    The homenet's Gplus ID

      +          The string concatenation operation
```

Once generated, the prefix is delegated to the homenet and /64 are assigned as specified in [I-D.arkko-homenet-prefix-assignment].

On the other ends of tunnels, the ULA prefix for this homenet is retrieved from the Network Settings post in Gplus and advertised through the connected homenets by injecting AS-External-LSAs in OSPFv3.

In case there are other ULA prefixes assigned in the homenet, they should also be advertised and routed to the connected homenets. Otherwise the Default Address Selection mechanism for IPv6 specified in [RFC3484] will lead to an unpredictable behaviour as the source addressed chosen by a host to communicate over the tunnel might not be in the prefix advertised on Gplus and then would not be routed at the other end. But having other ULA prefixes is non-desirable since it increases the odds of prefix collision. In our implementation, we assume there is no other ULA prefix assigned in the homenet.

Though we strive to avoid collisions while generating the ULA prefixes, the current design assumes there is no collision and does not treat such a case. Collisions might appear in two situations: either a Gplus router chooses the same prefix as one of its connected routers, or a Gplus router has two connected routers that have the same prefix. The best solution for this is left for further study.

5. Sharing services within your set of connected homes

Connecting homenets would be pointless without any service discovery mechanism. The aim is to allow a host to query services in connected homenets, and to let only the authorized services appear in the responses.

Inside a single home, automatic service discovery is enabled by the hybrid DNS-SD proxy mechanism specified in [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]. The following design assumes this running on all routers of the homenet and mostly relies on it to enable service discovery over multiple homes.

Connected homenets must have distinct domain names. Each homenet must either have a domain name that is owned by their administrator or generate a local one. In case of automatic generation we again have a problem of collisions and use Gplus IDs to make them the most unlikely possible. In order to make to it a minimum human-friendly too, the formatted display name of the associated Gplus page is put at the beginning, concatenated with an hyphen, the 10 last digits of the Gplus ID and the TLD. A generic TLD for homes might be defined in the future, though for this experiment we use ".test.".

To advertise this domain name across the homenet, the Gplus router advertises a Domain Name TLV.

To make hosts browse other homenets zones, a DNS Delegated Zone TLV must be advertise for each one of them. The S bit must be set to 0 because those zones are not full DNS-SD domains, and the B bit set to 1 so that they are recommended for browsing at b._dns-sd._udp.(domain). For each one, the domain name and authoritative DNS server address (a ULA address of the Gplus router) are retrieved from the Network Settings post published in Gplus.

Thus, the Gplus router’s DNS server receives from other homes all DNS-SD queries for its home’s domain name. Responses are filtered based on the source ULA address and the services authorized to the corresponding home. Notice also that A records and AAAA records that do not point to ULA addresses are dropped. A service is authorized if and only if a policy of one of the circles in which this home is allows it. For this experiment, a policy is defined as an authorized DNS-SD type of service (e.g. _http._tcp) associated to a circle, but finer granularity might be implemented (which adds complexity because of hosts changing DNS zones or name).

6. Security Considerations

The necessity to setup tunnels between homes is the first security consideration. Performing filtered DNS-SD service discovery and direct communication between homes might be possible thanks to end-to-end reachability enabled by IPv6. However DNS requests and most common services’ connections are not encrypted, which makes the enforcement of a secure channel between homes absolutely compulsory. The goal of the experiment is to allow homes to reach one another more easily than reaching the whole of the internet. Doing so, the boundaries of the homenet are redrawn to include mutliple homes, which brings up security issues. DNS requests and most common services’ connections are not encrypted, which motivates the enforcement of a secure channel between homes. Besides, tunnels also provide identity of the incoming packets.

Injecting global prefixes in other homes might be a way to advertise larger prefixes than those actually owned (e.g. advertising a /48 while only having a /56). Of course we could limit the size of advertised prefixes but this is not enough. One could imagine a PKI verification system but this would assume support from ISPs which is not currently offered. Using ULA prefixes mitigates this issue though it adds some others (already described in Section 4.2).

Still, defining firewall rules is probably the toughest security concern. First, to prevent spoofing, only packets with source and destination addresses in the expected ULA prefixes are allowed. Even though the firewall of OpenVPN servers is not able to know for sure which connected client has sent a packet as an IP address might be spoofed, potential harm is very limited because it will not receive any packet back.
Second, relying on unability to discover unauthorized services via DNS-SD is not sufficient, hence the need to accept only traffic corresponding to authorized services. This is a non-trivial general issue since a service cannot be reduced to a contact port and IP address tuple. This issue is left for further study.

7. Experimental results

TBD

8. IANA Considerations

This document contains no request to IANA.

9. Acknowledgements

The author would like to thank Mark Townsley, Alain Fiocco, Ole Troan and Markus Stenberg for valuable mentoring of the project, as well as Pierre-Alain Dupont, Nicolas Iooss, Maico Le Pape and Guillaume Mulocher for high contribution in the design and implementation of the prototype.

10. References

10.1. Normative References

[I-D.acee-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-acee-ospf-ospfv3-autoconfig-03 (work in progress), July 2012.

[I-D.arkko-homenet-prefix-assignment]
Arkko, J., Lindem, A., and B. Paterson, "Prefix Assignment in a Home Network", draft-arkko-homenet-prefix-assignment-04 (work in progress), May 2013.

[I-D.cheshire-mdnsext-hybrid]
Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-cheshire-mdnsext-hybrid-00 (work in progress), January 2013.

[I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "Home Networking Architecture for IPv6", draft-ietf-homenet-arch-08 (work in progress), May 2013.

[I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]
Stenberg, M., "Hybrid Unicast/Multicast DNS-Based Service Discovery Auto-Configuration Using OSPFv3", draft-stenberg-homenet-dnssdext-hybrid-proxy-ospf-00 (work in progress), June 2013.

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

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012.

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

10.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

Author’s Address

Cedric Dessez
Cisco Systems
Paris,
France

Email: cedric@dessez.fr