VIPR
Internet-Draft
Intended status: Standards Track
Expires: January 11, 2012

J. Rosenberg
jdrosen.net
C. Jennings
Cisco
M. Petit-Huguenin
Stonyfish

July 10, 2011

A Usage of Resource Location and Discovery (RELOAD) for Public Switched Telephone Network (PSTN) Verification
draft-petithuguenin-vipr-reload-usage-02

Abstract

Verification Involving PSTN Reachability (VIPR) is a technique for inter-domain SIP federation. VIPR makes use of the RELOAD protocol to store unverified mappings from phone numbers to RELOAD nodes, with whom a validation process can be run. This document defines the usage of RELOAD for this purpose.

Legal

This documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document relies heavily on the concepts and terminology defined in [VIPR-OVERVIEW] and will not make sense if you have not read that document first. As it defines a usage for RELOAD [P2PSIP-BASE], it assumes the reader is also familiar with that specification.

2. Terminology

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

3. Registering an E.164 number

To register an E.164 number a VIPR server stores a ViprRegistration structure using the fully qualified E.164 based number without any non-digit characters but the '+' character as Resource Name. For redundancy purpose, the VIPR server MUST store the same ViprRegistration structure two more times by using the same Resource Name prepended with the "COPY1" and "COPY2" character string respectively.

The contents of the ViprRegistration structure are as follow:

```
enum { reserved(0), node_id(8200), (65535) } ViprRegistrationType;

struct {
    select (ViprRegistrationType) {
      case node_id:
        NodeId            pvp_provider;

      /* This type can be extended */
    } ViprRegistrationData;

struct {
    ViprRegistrationType  type;
    uint16                length;
    ViprRegistrationData  data;
} ViprRegistration;
```

The ViprRegistration structure contains the following values:

type
The type of the registration

length
The length of the data structure, i.e. not counting the type and length fields

pvp_provider
The Node-ID of the peer to which an AppAttach request should be sent to initiate the PVP protocol

The Node-ID used in the pvp_provider field and in the key MUST be ready to process AppAttach requests for Application-ID 8470 at the time the registration is done.

VIPR supports multiple registrations for a single E.164 number by using a Dictionary Data Model. The dictionary key is the concatenation of the Node-ID and the VServiceId, resulting in a 24 bytes long value. Using the Node-ID of the node performing the store segments the keyspace of the dictionary so that no two peers ever store using the same key. Using the VService allows for a single VIPR server to service multiple clusters, and to ensure that numbers published by one cluster (using one VServiceID) do not clobber or step on numbers published by another cluster (using a different VServiceID).

The Store operations are paced into the overlay at a fixed rate. The VIPR server maintains a queue that is filled with store requests. The VIPR server services that queue at a fixed, provisioned rate, which is stored in a kind configuration variable named <store-rate-limit>.

4. Fetching a registration

A VIPR server wishing to validate a E.164 number will start 3 Fetch transactions using respectivelly the fully qualified E.164 based number without any non-digit characters but the ’+’ character as Resource Name, the same Resource Name prepended with the "COPY1" character string and finally the same Resource Name prepended with the "COPY2" character string.

The VIPR server will then inspects the elements in the 3 dictionary returned and will keep only the registrations that have the same key in at least 2 of the 3 dictionary returned. For each registration kept, the VIPR server will fetch the certificates associated with the Node-ID in the key using the CERTIFICATE_BY_NODE usage and will verify that the signature of the registration is valid.

The VIPR server can then send an AppAttach to the Node-ID found in the key and registration, using the Application-ID 8470. After the connection is established, the VIPR server can start PVP as specified in [VIPR-PVP].

5. VIPR-REGISTRATION Kind Definition

Name VIPR-REGISTRATION

Kind Ids The Resource Name for the VIPR-REGISTRATION Kind-ID is a fully qualified E.164 based number without any non-digit characters but the ’+’ character, prepended by either an empty character string, the "COPY1" character string or the "COPY2" character string. The data stored is a ViprRegistration, which contains the Node-ID of the peer to which an AppAttach request should be sent to initiate the PVP protocol.

Data Model The data model for the VIPR Kind-ID is dictionary. The dictionary key is the concatenation of the Node-ID and the VServiceId, resulting in a 24 bytes long value. Using the Node-ID of the node performing the store segments the keyspace of the dictionary so that no two peers ever store using the same key and using the VService allows for a single VIPR server to service multiple clusters, and to ensure that numbers published by one cluster (using one VServiceID) do not clobber or step on numbers published by another cluster (using a different VServiceID).

Access Control The VIPR-MATCH policy can only be used with a VIPR-REGISTRATION Kind-ID. In this policy, a given value MUST be written (or overwritten) if and only if the Node-ID in the pvp_provider field of the ViprRegistration structure is equal to the first 16 bytes of the dictionary key and if the same Node-ID is the one indicated in the SignerIdentity of the value. Note that VIPR always let the values stored expire, so the exists field is always equal to TRUE.

Quota This kind MUST use the proportional quota algorithm described in [RELOAD-QUOTA] by adding the <max-count-per> element with a value of "SIGNER" to the configuration file. The <max-count> value, which measures the amount of E.164 numbers a particular node can store, MUST be adjusted to account for the application-layer copies (COPY1 and COPY2). A VIPR server MUST provide enough Node-IDs to store all the E.164 numbers it is responsible for by dividing this count by the <max-count> value, itself adjusted by an additional 3x factor to make sure that the probability is low that a rejection occurs due to imperfect distribution of Resource-IDs across the ring.

[[Open Issue: need to adjust the multiplier - basically birthday problem!]]

The method for merging data after a partition follows the normal RELOAD rules around temporal ordering.

6. Overlay Configuration Document Extension

This document extends the overlay configuration document by defining a new element in the "urn:ietf:params:xml:ns:p2p:vipr" namespace.

The <store-rate-limit> defines the maximum rate in seconds that a VIPR server must use to execute Store requests.

The Compact Relax NG Grammar for this element is:

```
namespace vipr = "urn:ietf:params:xml:ns:p2p:vipr"

kind-parameter &= element vipr:store-rate-limit { xsd:unsignedInt }
```

7. Security Considerations

TBD

8. IANA Considerations

8.1. Access Control Policies

This document adds a new access control policy to the "RELOAD Access Control Policy" Registry:

Access-Policy	RFC
VIPR-MATCH	This document

This access control policy was described in Section 5.

8.2. Application-ID

This document adds a new application ID to the "RELOAD Application-ID" Registry:

Application	Application-ID	Specification
PVP	8470	This document

This access control policy was described in Section 5.

8.3. Data Kind-ID

This document adds a new Data Kind-ID to the "RELOAD Data Kind-ID Registry":

KIND	Kind-ID	RFC
VIPR-REGISTRATION	17	This document

This Kind-ID was defined in Section 5.

8.4. IETF XML Namespaces Registry

This document adds the following URN to the "XML Namespaces" class of the "IETF XML Registry":

urn:ietf:params:xml:ns:p2p:vipr

9. References

9.1. Normative References

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

[P2PSIP-BASE] Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-16 (work in progress), July 2011.

[VIPR-OVERVIEW]

Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-jennings-vipr-overview-00 (work in progress), April 2011.

[VIPR-PVP]
Rosenberg, J., Jennings, C., and M. Petit-Huguenin, "The Public Switched Telephone Network (PSTN) Validation Protocol (PVP)", draft-petithuguenin-vipr-pvp-01 (work in progress), June 2011.

[RELOAD-QUOTA]
Rosenberg, J., Jennings, C., and M. Petit-Huguenin, "Proportional Quota in REsource LOcation And Discovery (RELOAD)", draft-petithuguenin-p2psip-proportional-quota-01 (work in progress), July 2011.

9.2. Informative References

[ACCESS-CONTROL]
Petit-Huguenin, M., "Configuration of Access Control Policy in REsource LOcation And Discovery (RELOAD) Base Protocol", draft-petithuguenin-p2psip-access-control-03 (work in progress), July 2011.

Appendix A. Examples

The Resource Names and Resource-IDs for the E.164 number +1 408 555 5432 are:

Resource Name	Resource-ID
+14085555432	6abaec4308294521e2f600ab5fd01e5
COPY1+14085555432	9038006a8de78f818d318b60ed149d9
COPY2+14085555432	3d288c777bcf3aad38b077355026718

The VIPR-MATCH access control can be implemented with the following code (Using the notation in [ACCESS-CONTROL]):

```
   var equals = function(a, b) {
     if (a.length !== b.length) return false;
     for (var i = 0; i < a.length; i++) {
       if (a[i] !== b[i]) return false;
     }
     return true;
   };
   var length = configuration.node_id_length;
   return equals(entry.key.slice(0, length),
     entry.value.slice(4, length + 4))
     && equals(entry.key.slice(0, length), signature.node_id);
```

Appendix B. Release notes

This section must be removed before publication as an RFC.

B.1. Modifications between vipr-02 and vipr-01

- Made clear in the access control policy that exists is always equal to TRUE.
- Updated with new version of proportional-quota.
- The access control code now uses the configuration parameter.
- Assigned values to Application-ID and Kind-ID.
- Added running code section.
- Nits

B.2. Modifications between vipr-01 and vipr-00

- Moved most of the quota algorithm to a separate I-D named draft-petithuguenin-p2psip-proportional-quota.
- Removed the text saying that the same DHT can also be used for a RELOAD SIP usage, as it contradicts text in overview. Also the quota algorithm does not work with clients, but SIP registration uses clients.
- Added Terminology section
- Converted the TLV value stored to a structure using the syntax described in p2psip-base to not be dependent on VAP. The new structure is bit compatible with the old definition.
- Changed the dictionary key to be binary based instead of text based.
- Copied text from VAP explaining that the Store operations are queued and that the rate is limited.
- Added voting algorithm when fetch returns different results for the 3 copies.
- Added explanation on how to verify the signatures.

- Added text on how to form the PVP connection
- Rewrote some of the text so it looks more like a regular RELOAD usage.
- Removed section 3 "PeerID Shim" now that support for multiple Node-ID in certificates in fully integrated in RELOAD base.
- Filled IANA section
- Added examples of conversion from E.164 number to Resource-ID
- Added example code for the VIPR access control

B.3. Modifications between vipr-00 and dispatch-03

- Moved to new Working Group.

B.4. Modifications between dispatch-03 and dispatch-02

- Nits.
- Shorter I-Ds references.
- Fixed the peerID and VServiceID to be hexadecimal.
- Fixed the description of the dictionary entry
- Fixed the description of the TLV.
- Used +1 408 555 prefix for phone numbers in examples.
- Replaced peerId by Node-ID
- Replaced resourceID by Resource-ID

B.5. Running Code Considerations

- Reference Implementation for the kind and access control policy (<http://debian.implementers.org/testing/source/reload.tar.gz>). Marc Petit-Huguenin. Implements version -02.

Authors’ Addresses

Jonathan Rosenberg
jdrosen.net
Monmouth, NJ
US

Email: jdrosen@jdrosen.net
URI: http://www.jdrosen.net

Cullen Jennings
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1 408 421-9990
Email: fluffy@cisco.com

Marc Petit-Huguenin
Stonyfish

Email: marc@stonyfish.com