dispatch
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

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

October 25, 2010

A Usage of Resource Location and Discovery (RELOAD) for Public Switched Telephone Network (PSTN) Verification
draft-rosenberg-dispatch-vipr-reload-usage-03

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.

Table of Contents

1. Introduction 3
2. ViPR Usage 3
3. PeerID Shim 5
4. Security Considerations 6
5. IANA Considerations 6
6. References 6
6.1. Normative References 6
6.2. Informative References 6
Appendix A. Release notes 6
A.1. Modifications between rosenberg-03 and rosenberg-02 7
Authors' Addresses 7

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. The same DHT can also be used for a RELOAD SIP usage [P2PSIP-SIP].

2. ViPR Usage

The ViPR usage defines details for how the DHT is used for ViPR operations.

The ViPR usage defines Kind-ID 0x00000001. This Kind-ID is a dictionary entry. Its Resource-ID is defined through a transformation which takes an E.164 based number, and computes a Resource-ID as the least significant 128 bits of the SHA1 hash of the following string: Cat(CHOICE(null, "COPY", "COPY2"), number) That is, the Resource-ID is the hash of a string which is the concatenation of the number, prefixed with nothing, or the words "COPY1" or "COPY2".

For example, for number +14085555432:

Resource-ID = least128(SHA1("+14085555432"))

or

Resource-ID = least128(SHA1("COPY1+14085555432"))

or

Resource-ID = least128(SHA1("COPY2+14085555432"))

The object stored at this resource ID is a dictionary entry, which has a key and a value:

Object = {key,value}

Here, the key is formed by taking the Node-ID of the storing node in hex format, without the "0x", appending a "+", followed by the VServiceID in hex format, without the "0x". For example, if a peer with Node-ID

0x8f60f5eab753037e64ab6c53947fd532

receives a Publish with a VServiceID of

0x7eeb6a7036478351

The resulting key is:

8f60f5eab753037e64ab6c53947fd532+7eeb6a7036478351

Both parts of this key are important. Using the Node-ID of the node performing the store basically segments the keyspace of the dictionary so that no two peers ever store using the same key. Indeed, the responsible node will verify the signature over the stored data and check the Node-ID against the value of the key, to make sure that a conflict does not take place. The usage of 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 responsible node does not verify or check the VServiceID.

When a node receives a Store operation for this usage, the data itself has a signature. The node responsible for storing the data must verify this signature; the certificate will always be included in the data and indicate which Node-ID is used. The responsible node must check that this Node-ID is included in the cert. If the signature verifies, the responsible node checks that the data model is a dictionary entry. The key must meet the format above. The responsible node must check that it is a 32 character sequence of numbers and letters a-f, followed by a +, followed by a 16 character sequence of numbers and letters a-f. If this checks, the key is split in half along the plus. The first 32 characters are considered a hex value and compared with the Node-ID used for the signature. If they match, it is good. Otherwise the Store operation is rejected. If they did match, next the responsible node checks the value. It must be a TLV, with the same format used by VAP, and it must contain a single Node-ID attribute. The Node-ID must match that used for the signature. If they don’t match, the Store operation is rejected. If they do match, the next step is a quota check.

For each peer that the responsible node is storing data for, it must maintain a count of the number of unique dictionary entries being stored for that Node-ID. For each resourceID, each key constitutes a unique dictionary entry. So if a peer is storing 5 resourceIDs, and at each of those 5, there are two keys whose first 32 bits correspond to a particular Node-ID, it means this node is currently storing 10 unique dictionary entries for that Node-ID.

It takes the StorageQuota configuration parameter for this DHT, which measures the amount of numbers a particular node can store. That value is multiplied by nine (a 3x factor to account for the

application-layer copies (COPY1 and COPY2), and another 3x factor for replicas). Then, an addition 3x factor is added for rounding to make sure that the probability is low that a rejection occurs due to imperfect distribution of resourceIDs across the ring. (Open Issue: need to adjust this multiplier - basically birthday problem!) and then divided by the fraction of the hashspace owned by this ViPR server. If the result is less than one, it is rounded up to two. This is the max number of unique entries that can be stored for this storing peer ID. If the ViPR server is not yet storing this many entries for that peer ID, the store is allowed.

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

3. PeerID Shim

Because the ViPR implementation of RELOAD protocol makes use of the concept of multiple Node-ID on the same physical box, utilizing a single cert, the TLS handshakes alone are not sufficient to determine the entity on both sides of the TLS connection. As such, we will have a small "shim" type of protocol, which runs after TLS, but is not formally part of RELOAD.

When a node initiates a TLS connection towards another node, after the TLS completes, it sends this message. The message contains the Node-ID associated with this connection. The recipient gets this, and sends back a similar message, containing its Node-ID. Both sides will verify that, the Node-ID sent by the other side, are amongst the Node-IDs listed in the certificate. The connections are then stored in the connection tables, indexed by this Node-ID.

Furthermore, if, after this exchange, a node determines that it already has a connection in its connection table with that Node-ID on the far side, the older connection is closed. This is actually a critical security function! Without this, a user could clone ViPR servers utilizing the same certs, and each one can join the network.

Finally, once the exchange has taken place, the node compares the Node-ID from its peer with the current set of blacklisted Node-ID from the ACL that is distributed through the DHT. If the remote Node-ID appears on the list, the node closes the TCP/TLS connection immediately.

The reason we are using a non-reload message for this, is that we need to be 100% sure that this never propagates. It is strictly over a single connection and should never be routed. Indeed, had we not had this idea of multiple Node-ID in a single cert, this would have

effectively been accomplished through TLS. Alternatively, there is a TLS command for telling the other side who I expect them to be; however this is not implemented in older versions of OpenSSL, and so our shim forms an alternative to that which can be run on top of OpenSSL.

4. Security Considerations

TBD

5. IANA Considerations

TBD. Need to register items in IANA registries created by RELOAD.

6. References

6.1. Normative References

[P2PSIP-BASE]
Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-11 (work in progress), October 2010.

[VIPR-OVERVIEW]
Rosenberg, J., Jennings, C., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: Requirements and Architecture Overview", draft-rosenberg-dispatch-vipr-overview-04 (work in progress), October 2010.

6.2. Informative References

[P2PSIP-SIP]
Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "A SIP Usage for RELOAD", draft-ietf-p2psip-sip-05 (work in progress), July 2010.

Appendix A. Release notes

This section must be removed before publication as an RFC.

A.1. Modifications between rosenberg-03 and rosenberg-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

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