VIPR WG
Internet-Draft
Intended status: Informational
Expires: June 12, 2014

M. Barnes, Ed.
Polycom
C. Jennings
Cisco
J. Rosenberg
jdrosen.net
M. Petit-Huguenin
Unaffiliated

December 09, 2013

Verification Involving PSTN Reachability: Requirements and Architecture Overview

draft-jennings-vipr-overview-06

Abstract

The Session Initiation Protocol (SIP) has seen widespread deployment within individual domains, typically supporting voice and video communications. Though it was designed from the outset to support inter-domain federation over the public Internet, such federation has not materialized. The primary reasons for this are the complexities of inter-domain phone number routing and concerns over security. This document reviews this problem space, outlines requirements, and then describes a model and technique for inter-domain federation with SIP involving the Public Switched Telephone Network (PSTN), called Verification Involving PSTN Reachability (VIPR). VIPR addresses the problems that have prevented inter-domain federation over the Internet. It provides fully distributed inter-domain routing for phone numbers, authorized mappings from phone numbers to domains, a new technique for automated SIP anti-spam, and privacy of number ownership, all while preserving the trapezoidal model of SIP.

Status of This Memo

This Internet-Draft will expire on June 12, 2014.

Copyright Notice

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP) was originally published as [RFC2543] in May of 1999. This was followed by subsequent publication of [RFC3261], which brought the protocol to sufficient maturity to enable large scale market adoption.

SIP has achieved large scale market adoption with hundreds of implementations, spanning consumer products, enterprise servers, and large scale carrier equipment. It carries billions and billions of minutes of calls, and has become the standard for interconnection between products from different vendors. If one measures success in deployment, then clearly SIP is a success.

SIP was designed from the ground up to enable communications between users in different domains, all over the public Internet. The intention was that real-time communications should be no different than email or the web, with the same any-to-any connectivity that has fueled the successes of those technologies. However, when SIP is used between domains, it is typically through private federation agreements. While such agreements are positive, they have typically been limited to voice, which has limited the use of video and the growth of advanced SIP features, thus preventing the innovation that SIP was expected to drive. Thus, the any-to-any Internet federation model envisioned by SIP has not materialized at scale.

This document introduces a technology, called Verification Involving PSTN Reachability (VIPR), that breaks down the barriers that have prevented inter-domain voice, video and other multimedia services. By stepping back and changing some of the most fundamental assumptions about federation, VIPR is able to address the key problems preventing its deployment. VIPR focuses on incremental deployability. At the same time, VIPR ensures that SIP's trapezoidal model of direct federation between domains without any intermediate processing beyond IP transport is realized. That model is required in order to allow innovative new services to be deployed.

Despite the advantages of the VIPR system, its open, peer-to-peer character makes it vulnerable to certain security and privacy vulnerabilities (see especially Section 7.5). After consideration of potential countermeasures, the VIPR working group elected not to pursue VIPR for standardization. This document therefore describes VIPR for informational purposes, as VIPR has seen some field

deployment, and it is furthermore believed that the techniques utilized by VIPR might be reused in new standard architectures in the future.

2. Conventions and Terminology

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

Call Agent: An entity in a SIP enabled domain that supports VIPR. The Call Agent performs call processing on behalf of one or more user agents represented by E.164 numbers within the domain.

Ticket: A shared secret that is generated after a PSTN call to enable secure call setup on a subsequent inter-domain IP call enabled by VIPR.

User Agent: As defined in [RFC3261], with the restriction that the user agent must have an associated E.164 number.

3. Problem Statement

The first question that must be asked is this - why haven't we seen widespread adoption of inter-domain SIP federation? The reason for this is due to problems with the following - summarized in order of importance/impact:

1. Phone number routing

2. Open pinhole

3. Quality of service

4. Troubleshooting

The first two are the most significant.

3.1. The Phone Number Routing Problem

Inter-domain federation requires that the sending domain determine the address of the receiving domain, in the form of a DNS name (example.com) or one or more IP addresses that can be used to reach the domain. In email and in the web, this is easy. The identifiers used by those services - the email address and web URL respectively - embed the address of the receiving domain. A simple DNS lookup is all that is required to route the connection. SIP was designed to use the same email-style identifiers.

However, most SIP deployments utilize phone numbers in the form of E.164 numbers [E.164], and not email-style SIP URIs. This is due to the huge installed base of users that continue to exist solely on the PSTN. In order to be reached by users on the PSTN, and in order to reach them, users in SIP deployments need to be assigned a PSTN phone number. Users in SIP deployments need to place that phone number on business cards, use it in their email signatures, and in general, give it out to their friends and colleagues, in order to be reached. While those users could additionally have an email style SIP URI, the phone number serves as a single, global identifier that works for receiving calls from users on the PSTN as well as users within the same SIP domain.

There are several reasons why two identifiers are used when one will suffice. The universality of PSTN phone numbers is the reason why most SIP deployments continue to use them - often exclusively.

Another reason is that many SIP deployments utilize hardphones or telephony adaptors, and the user interfaces on these devices - patterned after existing phones - only allow phone number based dialing. Consequently, these users are only allocated PSTN phone numbers, and not email-style SIP URI.

Finally, a large number of SIP deployments are in domains where the endpoints are not IP. Rather, they are circuit based devices, connected to a SIP network through a gateway. SIP is used within the core of the network, providing lower cost transit, or providing add-on services. Clearly, in these deployments, only phone numbers are used.

Consequently, to make inter-domain federation incrementally deployable and widely applicable, it needs to work with PSTN phone numbers rather than email-style SIP URIs. Telephone numbers, unlike email addresses, do not provide any indication of the address of the domain which "owns" the phone number. Indeed, the notion of phone number ownership is somewhat cloudy. Phone numbers can be ported between carriers. They can be assigned to a user or enterprise, and then later re-assigned to someone else. Phone numbers are granted to users and enterprises through a complex delegation process involving

the ITU, governments, and telecommunications carriers, often involving local regulations that vary from country to country.

Therefore, in order to deploy inter-domain federation, domains are required to utilize some kind of mechanism to map phone numbers to the address of the domain to which calls should be routed. Though several techniques have been developed to address this issue, none have achieved large-scale Internet deployments.

3.2. The Open Pinhole Problem

The inter-domain federation mechanism built into SIP borrows heavily from email. Each domain runs a SIP server on an open port. When one domain wishes to contact another, it looks up the domain name in the DNS, and connects to that server on the open port. Here, "open" means that the server is reachable from anywhere on the public Internet, and is not blocked by firewalls.

This simple design worked well in the early days of email. However, the email system has now become plagued with spam. This has resulted in administrators spending a significant amount of time maintaining spam filters. This does not always benefit the end users as in some cases valid emails are dropped without the user being notified. Thus, administrators of SIP domains are rightfully concerned that if they make a SIP server available for anyone on the Internet to contact, it will open the floodgates for SIP spam, which is far more disruptive than email-based spam [RFC5039]. Administrators are also concerned that an open server will create a back-door for denial-of-service and other attacks that can potentially disrupt their voice and video services. Administrators are often not willing to take that risk since voice deployments demand higher uptimes and better levels of reliability than email, especially for enterprises.

Fears around spam and denial-of-service attacks, when put together, form the "open pinhole problem" - that domains are not willing to enable SIP on an open port facing the Internet.

To fix this, a new model for federation is needed - a model where these problems are addressed as part of the fundamental design rather than after the functionality has been deployed.

3.3. Quality of Service Problem

The Internet does not provide any Quality of Service (QoS) guarantees. All traffic is best effort. This is not an issue for data transaction services, like web and email. It is, however, a concern when using real-time services, such as voice and video.

That said, there are a large number of existing SIP deployments that run over the Internet. Though the lack of QoS is a concern, it has not proven a barrier to deployment. It is believed that if if the more fundamental issues - the phone number routing and open pinhole problems - can be addressed, the QoS problem will be a non-issue. As such, QoS is not discussed further in this or other VIPR specifications.

3.4. Troubleshooting Problem

The final problem that is prohibiting large scale inter-domain federation is troubleshooting. When connecting calls between domains, problems can occur. Calls can be blocked. Calls can be misdelivered. Features sometimes don't work. There can be one-way media or no media at all. The video may not start. Call quality can be poor.

These problems are common in SIP deployments, and they are tough to troubleshoot even within a single administrative domain. When real-time services extend inter-domain, the problem becomes worse.

Fortunately, some work has been completed to improve the ability for network administrators to diagnose SIP problems. A Common log format [RFC6873] has been developed. Other work underway, such as consistent session IDs [I-D.ietf-insipid-session-id-reqts] and [I-D.jones-insipid-session-id] can help troubleshoot interdomain calls.

In addition to the above, any new technology that facilitates inter-domain federation needs to have troubleshooting built-in, so that it is not a barrier to deployment. Further consideration of necessary built-in techniques for troubleshooting is required for successful deployment of VIPR.

4. Summary of Existing Solutions

Given the value of inter-domain SIP federation, there are existing deployed solutions summarized below. However, each solution approach has fundamental limitations that have inhibited widespread deployment.

4.1. Domain Routing

The first solution for SIP inter-domain federation is built into SIP itself - domain routing. In this technique, users utilize email-style SIP URIs as identifiers. By utilizing the DNS lookup mechanism defined in [RFC3263], SIP enables calls to be routed between domains in much the same way email is routed between domains.

This technique works well in theory, but it has two limitations which have limited its deployment:

1. The majority of SIP deployments utilize phone numbers, often exclusively. In such a case, domain routing cannot be used.

2. Domain federation brings with it the possibility (and strong likelihood) of the same levels of spam and DoS attacks that have plagued the email system.

These issues have already been discussed in sections Section 3.1 and Section 3.2 respectively.

4.2. Public ENUM

Public ENUM, defined in [RFC6116] addresses the phone number routing problem by placing phone numbers into the public DNS. Clients can then perform a simple DNS lookup on a phone number, and retrieve a SIP URI which can be used to route to that phone number.

Unfortunately, public ENUM requires that the entries placed into the DNS be populated following a chain of responsibility that mirrors the ownership of the numbers themselves. This means that, in order for a number to be placed into the DNS, authorization to do so must start with the ITU, and from there, move to the country, telecom regulator, and ultimately the end user. The number of layers of bureaucracy required to accomplish this is non-trivial. In addition, the telecom operators - that would be partly responsible for populating the numbers into the DNS - have little incentive to do so. As a consequence, public ENUM is largely empty, and is likely to remain so for the foreseeable future.

Instead, ENUM has evolved into a technique for federation amongst closed peering partners, called private ENUM or infrastructure ENUM [RFC5067]. While there is value in this technology, it does not enable the open federation that public ENUM was designed to solve.

4.3. Private Federations

Private federations are a cooperative formed amongst a small number of participating domains. The cooperative agrees to use a common technique for federation, and through it, is able to connect to each other. There are many such federations in use today.

Some of these federations rely on a central database, typically run by the federation provider, that can be queried by participating domains. The database contains mappings from phone numbers to domains, and is populated by each of the participating domains, often

manually. Each domain implements an agreed-upon query interface that can be used to access the database when a number is called. Sometimes ENUM is used for this interface (called private ENUM), other times, a SIP redirection is used. Some federations also utilize private IP networks in order to address QoS problems.

Private federations work, but they have one major limitation: scale. As the number of participating domains grows, several problems arise. Firstly, the size of the databases become difficult to manage. Secondly, the correctness of the database becomes an issue, since the odds of misconfigured numbers (either intentionally or accidentally) increases. As the membership grows further, the odds increase that malicious domains will be let in, introducing a source of spam and further problems. The owner of the federation can - and often does - assume responsibility for this, and can attempt to identify and shut down misbehaving participants. Indeed, as the size of the federations grow, the owner of the federation needs to spend increasing levels of capital on maintaining it. This often results in the owners charging for membership, which can be a barrier to entry.

5. Key Requirements

From the discussion on the problems of inter-domain federation and the solutions that have been attempted so far, several key requirements emerge:

REQ-1: The solution must allow for federation between any number of domains.

REQ-2: The solution must enable users in one domain to identify users in another domain through the use of their existing E.164 based phone numbers.

REQ-3: The solution must work with deployments that utilize any kind of endpoint, including non-IP phones connected through gateways, IP softphones and hardphones.

REQ-4: The solution must not require any change in user behavior. The devices and techniques that users have been using previously to make inter-domain calls must continue to work, but now result in inter-domain calls using IP.

REQ-5: The solution must work worldwide, for any domain anywhere.

REQ-6: The solution must not require any new services from any kind of centralized provider. A domain should be able to deploy equipment and connect to the federation without any interaction with or authorization from a centralized provider.

REQ-7: The solution must not require any prior arrangement between domains in order to facilitate federation between those domains. Federation must occur opportunistically - connections established when they can be.

REQ-8: The solution must work for domains of any size - starting with a single phone up to the largest telecom operator with tens of millions of numbers.

REQ-9: The solution must have built-in mechanisms for preventing spam and DoS attacks. These mechanisms must be fully automated.

REQ-10: The solution must not require any processing whatsoever by SIP or RTP intermediaries. It must be possible for a direct SIP connection to be established between participating domains.

REQ-11: The solution should adapt to VIPR call failures. The solution should allow the user to make calls using the inter-domain calling mechanism used prior to the initial VIPR-enabled call.

6. Executive Overview

Verification Involving PSTN Reachability (VIPR) is aimed at solving the problems that have prevented large-scale Internet-based SIP federation of voice and video. VIPR solves these problems by creating a hybrid of three technologies - the PSTN itself, a Peer to Peer (P2P) network, and SIP. By using these three technologies together, VIPR enables an incrementally deployable solution to federation.

6.1. Key Properties

VIPR has several important properties that enable it to solve the federation problem:

Works With Numbers: VIPR enables federation for existing PSTN phone numbers. It does not require users or administrators to know or configure email-style identifiers. It does not require the allocation of new numbers. It does not require a change in user behaviors.

Works with Existing Endpoints: VIPR does not require any changes to endpoints. Consequently, it works with existing SIP endpoints and with non-IP endpoints connected through gateways.

Verified Mappings: VIPR ensures that phone calls cannot be misrouted or numbers stolen. The biggest issue in mapping from a phone number to a domain or IP address, is determining whether the mapping is correct - i.e., does the domain really own the given phone number? While solutions like ENUM have solved this problem by relying on centralized delegations of authorization, VIPR provides a secure mapping in a fully distributed way.

Worldwide: VIPR works worldwide. Any domain that is connected to both the PSTN and the Internet can participate. Since VIPR does not depend on availability of any regional services beyond IP and PSTN access - both of which are already available globally - VIPR itself is globally available.

Scalibility: VIPR is scaleable. Any number of domains can participate.

Self-Scale: VIPR self-scales. This means that the amount of computation, memory, and bandwidth that a domain must deploy scales in direct proportion to the size of their own user base.

Self-Learning: VIPR is completely automated. A domain does not require configuration of any information about another domain. It does not require provisioning of IP addresses, domain names, certificates, phone number prefixes or routing rules.

Automated Anti-Spam VIPR has a built-in mechanism for preventing SIP spam, which is specific to SIP. It is fundamentally different from existing SIP anti-spam techniques which borrow from email [RFC5039]. This new technique is fully automated, and requires no configuration by administrators and no participation from end users.

Feature Velocity: VIPR enables direct SIP connections between two domains seeking to federate. There are no SIP intermediaries of any sort between the two. This means that domains have no dependencies on intermediaries for deployment of new features.

Secure: Security is a fundamental part of VIPR and cannot be disabled.

Reliable: VIPR is reliable. Through its hybridization of the PSTN and the Internet, it ensures that calls always go through, even in cases of network failure or limited IP connectivity.

In order to achieve a solution with these properties, past assumptions about how federations should work must be challenged.

6.2. Challenging Past Assumptions

Two unstated assumptions of SIP federation are challenged by VIPR.

The first assumption that federation solutions have made is this:

> The purpose of SIP federation is to eliminate the PSTN, and consequently, we cannot assume the PSTN itself as part of the solution.

Though unstated, this assumption has clearly been part of the design of existing solutions. SIP federation based on email-style URIs, as defined in RFC 3261, doesn't utilize nor make mention of the PSTN. Solutions like ENUM, or private registries, also do not utilize nor make mention of the PSTN. However, such approaches ignore an incremental solution - a solution which utilizes the PSTN itself to solve the hard problems in SIP federation.

There are many advantages to leveraging the PSTN. It reaches worldwide. It provides a global numbering translation service that maps phone numbers to circuits. It is highly reliable, and provides QoS. It has been built up over decades to achieve these goals. Thus, building upon rather than replacing the PSTN, can provide the necessary functionality once another assumption is challenged.

This second assumption is:

> A federation solution must be the same as the final target federation architecture, and not just a step towards it.

SIP's email-style federation was a pure 'target architecture'. ENUM was the same - a worldwide global DNS database with everyone's phone numbers providing open connectivity.

Historically, technologies are more successful when they are incrementally deployable. As such, VIPR is very much focused on incremental deployability. It discards the notion of perfect IP federation for a solution that federates most, but not all calls, by relying on the PSTN to fill in the gaps.

6.3. Technical Overview

A high level view of the VIPR architecture with an example is shown in Figure 1. The figure shows four different domains, example.com, example.net, example.org and example.edu, federated using VIPR technology. Each domain is connected to both the public Internet and to the traditional PSTN. For simplicity, the connection for the call agents in example.org and example.edu to the PSTN is not indicated in the diagram as that interface is not relevant to the subsequent examples.

```
                         +-------+    +-------+
                         |  Call |    |  Call |
          example.org    | Agent |    | Agent |  example.edu
                         |       |    |       |
                         +-------+    +-------+
                             \           /
                              \         /
                               \       /
                                \     /
                                   |
                             //--------\\
                           |//          \\|
                           |   Internet   |
             +-------+     |\\          //|    +-------+
             |  Call |------ \\ _______//------|  Call |
     //\\    | Agent |                         | Agent |    //\\
     \  /    |       |                         |       |    \  /
      \/  ---|       |      +-----------+      |       |---- \/
     User    |       |======|           |======|       |    User
    Agent    +-------+      |    PSTN   |      +-------+   Agent
             example.com    |           |    example.net
                            +-----------+
```

Figure 1: High Level Architecture

For purposes of explanation, it is easiest to think of each domain as having a single call agent which participates in the federation solution. The functionality is decomposed into several sub-components, and this is discussed in more detail below. The call agent is connected to one or more user agents in the domain, and is responsible for routing calls, handling features, and processing call state. The call agent is stateful, and is aware of when calls start and stop. Additional detail for the functional components of this architecture are provided in [I-D.petithuguenin-vipr-framework].

Assume that all four domains have a 'fresh' installation of VIPR, and that domain example.net 'owns' +1 408 555 5xxx, a block of 1000 numbers allocated by its PSTN provider.

The VIPR mechanism can be broken into four basic steps: storage of phone numbers, PSTN first call, validation and caching, and subsequent SIP call(s).

6.3.1. Storage of Phone Numbers

The first step is that the call agents form a single, worldwide P2P network, using a VIPR specific usage [I-D.petithuguenin-vipr-reload-usage] of RELOAD [I-D.ietf-p2psip-base] with a variant of the Chord algorithm. This P2P network forms a distributed hash table (DHT) running amongst all participating domains. A distributed hash table is like a simple database, allowing storage of key-value pairs, and lookup of objects by key. Unlike a normal hash table, which resides in the memory of a single computer, a distributed hash table is spread across all of the servers which make up the P2P network. In this case, it is spread across all of the domains participating in the VIPR federation.

The problem solved by the variant of the Chord algorithm (and by other DHT algorithms), is an answer to the following: given that the desired operation is to read or write an object with key K, which node in the DHT is the box that currently stores the object with that key? The P2P SIP variant of the Chord algorithm provides an algorithm which routes read and write operations through nodes in the DHT until they eventually arrive at the right place. With Chord, this will take no more than log2N hops, where N is the number of nodes in the DHT. Consequently, for a DHT with 1024 nodes, 10 hops are required in the worst case. For 2048, 11 hops. And so on. The logarithmic factor allows DHTs to achieve efficient scale and to provide a large amount of storage summed across all of the nodes that make up the DHT.

This logarithmic hopping behavior also means that each node in the DHT does not need to establish a TCP/TLS connection to every other node. Rather, connections are established to a smaller subset - just log(N) of the nodes.

In DHTs, each participating entity is identified by a Node-ID. The Node-ID is a 128 bit number, assigned randomly to each entity. They have no inherent semantic meaning; they are not like domain names or IP addresses.

In the case of VIPR, each call agent is identified by one or more Node-IDs. For purposes of discussion, consider the case where the

call agent has just one Node-ID. Each participating domain, including example.net in our example, uses the DHT to store a mapping from each phone number that it owns, to the domain's Node-ID. In the case of example.net, it would store 1000 entries into the DHT, each one being a mapping from one of its phone numbers, to the domain's Node-ID. Furthermore, when the mappings are stored, the mapping is actually from the SHA-1 hash of the phone number, to the Node-ID of the call agent which claims ownership of that number.

For example, if the Node-ID of the call agent in domain example.net is 0x1234 (a shorter 16 bit value to simplify discussion), the entries stored into the DHT by example.net would be:

Key	Value
SHA1(+14085555000)	0x1234
SHA1(+14085555001)	0x1234
SHA1(+14085555002)	0x1234
.....	
SHA1(+14085555999)	0x1234

Figure 2: DHT Contents

It is important to note that the DHT does not contain phone numbers (it contains hashes of them), nor does it contain IP addresses or domain names. Instead, it is a mapping from the hash of a phone number (in E.164 format) to a Node-ID.

example.net will store this mapping when it starts up, or when a new number is provisioned. The information is refreshed periodically by example.net. The actual server on which these mappings are stored depends on the variant of the Chord algorithm. Typically, the entries will be uniformly distributed amongst all of the call agents participating in the network.

6.3.2. PSTN First Call

At some point, a user agent (Alice) in example.com makes a call to +1 408 555 5432, which is her colleague Bob. Even though both sides have VIPR, the call takes place over the plain old PSTN, per Figure 3. Alice talks to Bob for a bit, and they hang up.

```
         +-------+                          +-------+
         |  Call |                          |  Call |
 //\\    | Agent |                          | Agent |    //\\
 \  /    |       |                          |       |    \  /
```

```
      \/  ---|       |      +-----------+      |       |---- \/
      Alice  |       |<=======<========>======>|       |    Bob
             +--------+     |    PSTN  |      +--------+
             example.com    |           |    example.net
                            +-----------+
```

Figure 3: PSTN First Call

At a random point in time after the call has completed, the call agent in example.com "wakes up" and says to itself, "that's interesting, someone in my domain called +1 408 555 5432, and it went over the PSTN. I wonder if that number is reachable over IP instead?". To make this determination, it hashes the called phone number, and looks it up in the DHT. It is important to note that this lookup is not at the time of an actual phone call - this lookup process happens outside of any phone call, and is a background process.

The query for +1 408 555 5432 will traverse the DHT, and eventually arrive at the node that is responsible for storing the mapping for that number. Typically, that node will not be example.net, but rather one of the other nodes in the network (e.g., example.org). In many cases, the called number will not find a matching mapping in the DHT. This happens when the number that was dialed is not owned by a domain participating in VIPR. When that happens, example.com takes no further action. Next time there is another call to the same number, it will repeat the process and check once more whether the dialed number is in the DHT.

In this case, there is a match in the DHT, and example.com learns the Node-ID of example.net. It then proceeds to the validation step per Section 6.3.3. It is also possible that there are multiple matches in the DHT. This can happen if another domain - example.edu for example - also claims ownership of that number. When there are multiple matching results, example.com learns all of them, and performs the validation step with each.

6.3.3. Validation and Caching

Why not just store the domain in the DHT, instead of the Node-ID? If the domain was stored in the DHT, once example.com performed the lookup, it would immediately learn that the number maps to example.net, and could then make a direct SIP call next time.

The main reason this doesn't work is security. The information in the DHT is completely untrusted. There is nothing so far that enables example.com to know that example.net does, in fact, own the phone number in question. Indeed, if multiple domains make a claim on the number, it has no way to know which one (if any) actually owns it.

To address this critical problem, VIPR requires a mechanism called phone number validation. Phone number validation is a key concept in VIPR. There are several models for this validation as detailed in [I-D.petithuguenin-vipr-pvp]. The essential idea is that example.com will connect to the example.net server, by asking the DHT to form a connection to example.net's Node-ID. Once connected, example.com demands proof of ownership of the phone number. This proof comes in the form of demonstrated knowledge of the previous PSTN call. When a call was placed from example.com to +1 408 555 5432, the details of that call - including its caller ID, start time, and stop time, create a shared secret referred to as a "ticket", - information that is only known to entities that participated in the call. Thus, to obtain proof that example.net really owns the number in question, example.com will demand a knowledge proof - that example.net is aware of the details of the call. A consequence of this is that the following property is maintained:

> A domain can only call a specific number over SIP, if it had previously called that exact same number over the PSTN.

This property is key in fighting spam and denial-of-service attacks. Because calling numbers on the PSTN costs money - especially international calls - VIPR creates a financial disincentive for spammers. For a spammer to ring every phone in a domain with a SIP call, it must have previously called every number in the domain with a PSTN call, and had a successfully completed call to each and every one of them. [I-D.petithuguenin-vipr-sip-antispam] provides an overview and further details on the security mechanisms for VIPR for mitigation of SPAM.

There are a great many details required for this validation protocol to be secured. For example, the mechanism needs to handle the fact that call start and stop times won't exactly match on both sides. It needs to deal with the fact that many calls start on the top of the hour. It needs to deal with the fact that caller ID is not often delivered, and when it is delivered, is not reliable. It needs to deal with the fact that example.com may in fact be the attacker, trying to use the validation protocol to extract the shared secret from example.net. All of this is, in fact, handled by the protocol. The protocol is based on the Secure Remote Password for TLS Authentication (SRP-TLS) [RFC5054], and is described more fully in [I-D.petithuguenin-vipr-pvp].

Towards the end of the validation process, domains example.com and example.net had determined that each was, in fact in possession of the shared secret information about the prior PSTN call. However, neither side has any information about the domain names of the other side.

At the end of the validation process, both example.com and example.net have been able to ascertain that the other side did in fact participate in the previous PSTN call. At that point, example.com sends its domain name to example.net as shown in Figure 4.

```
                         +-------+    +-------+
                         |  Call |    |  Call |
          example.org    | Agent |    | Agent |  example.edu
                         |       |    |       |
                         +-------+    +-------+
                            \             /
   +----------------------+  \           /
   | Hi, I am example.com.|   \         /
   | How do I reach you?  |    \       /
   +--------------\-------+  //-------\\
                   \        //         \\
                +===\======>========>========>=====+
                ^          |    Internet  |        |
                |          |              |        v
             +-------+     |\\          //|    +-------+
             |  Call |------ \\ _______//------|  Call |
     //\\    | Agent |                         | Agent |    //\\
     \  /    |       |                         |       |    \  /
      \/  ---|       |                         |       |---- \/
      Alice  |       |                         |       |    Bob
```

```
             +-------+                         +-------+
             example.com                      example.net
```

Figure 4: Ticket Validation Step 1

Next, the example.net domain generates the ticket. The ticket has three fundamental parts to it:

1. The phone number that was just validated - in this case, +1 408 555 5432.

2. The domain name that the originating side claims it has - example.com in this case.

3. A signature generated by example.net, using a key known to itself only, over the other two pieces of information.

Then, example.net sends to example.com - all over a secured channel - a SIP URI to use for routing calls to this number, and a ticket, as shown in Figure 5. The ticket is a cryptographic object, opaque to example.com, but used by example.net to allow incoming SIP calls. It is similar in concept to kerberos tickets - it is a grant of access. In this case, it is a grant of access for example.com to call +1 408 555 5432, and only +1 408 555 5432.

```
                         +-------+    +-------+
                         |  Call |    |  Call |
          example.org    | Agent |    | Agent |  example.edu
                         |       |    |       |
                         +-------+    +-------+
                            \             /
                             \           /    +------------------------+
                              \         /     | Here is your ticket    |
                               \       /      | & SIP URI to reach Bob |
                             //-------\\      +----/-------------------+
                            //         \\         /
                +==========<========<========<===/=+
                |          |    Internet  |        ^
                v          |              |        |
             +-------+     |\\          //|    +-------+
             |  Call |------ \\ _______//------|  Call |
     //\\    | Agent |                         | Agent |    //\\
     \  /    |       |                         |       |    \  /
```

```
      \/  ---|       |                         |       |---- \/
      Alice  |       |                         |       |    Bob
             +-------+                         +-------+
             example.com                       example.net
```

Figure 5: Ticket Validation Step 2

The example.com call agent receives the SIP URI and ticket, and stores both of them in an internal cache. This cache builds up slowly over time, containing the phone number, SIP URI, and ticket, for those numbers which are called by example.com and validated using VIPR. Because the cache entries are only built for numbers which have actually been called by users in the enterprise, the size of the cache self-scales. A call agent supporting only ten users will build up a cache proportional to the volume of numbers called by ten people, whereas a call agent supporting ten thousand users will build up a cache which is typically a thousand times larger.

This cache, containing the phone number, SIP URI and ticket will be accessed later when Alice (or another caller from the same call agent) makes another call to Bob, as detailed in Section 6.3.4.

6.3.4. SIP Call

At some point in the future, another call is made to +1 408 555 5432. The caller could be Alice, or it could be any other user attached to the same call agent. This time, the call agent notes that it has a cached entry (including the SIP URI and ticket) for the number in question. It is possible that there are multiple entries for a given number. For example, both an Enterprise and Service Provider may register the same number in the RELOAD distributed database. It may also be possible to fork a call using the multiple entries . [Editor's note: this requires further discussion as to whether we want to allow multiple entries.]

The example.com call agent attempts to contact the SIP URI by establishing a TCP/TLS connection to the SIP URI it learned. If a connection cannot be made and there are no other cached entries for the number in question, the call agent proceeds with the call over the PSTN. This ensures that, in the event of an Internet failure or server failure, the call can still proceed. Assuming the connection is established, the example.com call agent sends a SIP INVITE to the terminating call agent, over this newly formed secure connection. The SIP INVITE request also contains the ticket, placed into a new SIP header field in the message.

When the SIP INVITE arrives at the example.net call agent, the call agent can extract the ticket from the new SIP header field. This ticket is an object, opaque to example.com, that was previously generated by the example.net call agent as described in Section 6.3.3. example.net first verifies the signature over the ticket. Remember that the example.net agent is the one that generated the ticket in the first place; as such, it is in possession of the key required to validate the signature. Once validated, it performs two checks:

1. It compares the phone number in the call setup request (the Request URI) against the phone number stored in the ticket.

2. It compares the domain name of the calling domain, learned from the certificates in the mutual TLS exchange, against the domain name stored in the ticket.

If both match, the example.net call agent knows that the calling party is in fact the domain they claimed previously, and that they had in fact gone through the validation process successfully for the number in question. At this time, the call is now completed per normal SIP processing.

7. Security Considerations

This section provides an overview of some of the key threats and how they are handled at a high level. Note that the detailed security solutions to handle the threats are detailed in the other relevant VIPR documents as referenced in the sections below.

7.1. Attacks on the DHT

Attackers could attempt to disrupt service through a variety of attacks on the DHT.

Firstly, it must be noted that the DHT is never used at call setup time. It is accessed as a background task, solely to learn NEW numbers and SIP URIs that are not already known. If an attacker was able to completely destroy the P2P network, it would not result in a single call to fail. Furthermore, it would not cause calls to revert to the PSTN - calls to SIP URIs learned previously would still go over the IP network. The only impact to such a devastating attack is that a domain could not learn SIP URIs for new numbers, until the DHT is restored to service. This service failure is hard for users and administrators to even notice.

That said, VIPR prevents many of these attacks. The DHT itself is secured using TLS - its usage is mandatory. Quota mechanisms are put

into place that prevent an attacker from storing large amounts of data in the DHT as described in [I-D.petithuguenin-vipr-proportional-quota]. Other attacks are prevented by mechanisms defined by RELOAD [I-D.ietf-p2psip-base] itself, and are not VIPR specific.

7.2. Theft of Phone Numbers

A key security threat that VIPR is trying to address is the theft of phone numbers. In particular, a malicious domain could store, in the DHT, phone numbers that it does not own, in an attempt to steal calls targeted to those numbers. This attack is prevented by the core validation mechanism as described in [I-D.petithuguenin-vipr-pvp] , which performs a proof of knowledge check to verify ownership of numbers.

An attacker could try to claim numbers it doesn't own, which are claimed legitimately by other domains in the VIPR network. This attack is prevented as well. Each domain storing information into the DHT can never overwrite information stored by another domain. As a consequence, if two domains claim the same number, two records are stored in the DHT. An originating domain will validate against both, and only one will validate - the real owner.

An attacker could actually own a phone number, use it for a while, validate with it, and build up a cache of routes at other domains. Then, it gives back the phone number to the PSTN provider, who allocates it to someone else. However, the attacker still claims ownership of the number, even though they no longer have it. This attack is prevented by expiring the learned routes after a while. Typically, operators do not re-assign a number for a few months, to allow out-of-service messages to be played to people that still have the old number. Thus, the TTL for cached routes is set to match the duration that carriers typically hold numbers.

An attacker could advertise a lot of numbers, most of which are correct, some of which are not. VIPR prevents this by requiring each number to be validated individually.

An attacker could make a call so they know the call details of the call they made and use this to forge a validation for that call. They could then try to convince other users, which would have to be in the same domain as the attacker, to trust this validation. This is mitigated by not sharing validations inside of domains where the users that can originate call from that domain are not trusted by the domain.

7.3. Spam

Another serious concern is that attackers may try to launch SIP spam (also known as SPIT) calls into a domain. As described in Section 6.3.3 and as detailed in [I-D.petithuguenin-vipr-sip-antispam], VIPR prevents this by requiring that a domain make a PSTN call to a number before it will allow a SIP call to be accepted to that same number. This provides a financial disincentive to spammers. The current relatively high cost of international calling, and the presence of national do-not-call regulations, have prevented spam on the PSTN to a large degree. VIPR applies those same protections to SIP connections.

VIPR still lowers the cost of communications, but it does so by amortizing that savings over a large number of calls. The costs of communications remain high for infrequent calls to many numbers, and become low for frequent calls to a smaller set of numbers. Since the former is more interesting to spammers, VIPR gears its cost incentives away from the spammers, and towards domains which collaborate frequently.

It is important to note that VIPR does not completely address the spam problem. A large spamming clearing house organization could actually incur the costs of launching the PSTN calls to numbers, and then, in turn, act as a conduit allowing other spammers to launch their calls to those numbers for a fee. The clearinghouse would actually need to transit the signaling traffic (or, divulge the private keys to their domain name), which would incur some cost. As such, while this is not an impossible situation, the barrier is set reasonably high to start with - high enough that it is likely to deter spammers until it becomes a highly attractive target, at which point other mechanisms can be brought to bear.

7.4. Eavesdropping

Another class of attacks involves outsiders attempting to listen in on the calls that run over the Internet, or obtain information about the call through observation of signaling.

All of these attacks are prevented by requiring the usage of SIP over TLS and SRTP. These are mandatory to use.

7.5. Privacy Leakage and Malicious Servers

A further form of attack involves adding malicious VIPR servers to a widely implemented (e.g., national or international) RELOAD overlay. This attack is specific to an uncontrolled RELOAD overlay, in which any individual or enterprise could add their own VIPR server to the overlay without authorization, verification or bias.

In this scenario, a malicious VIPR server could be used for analyzing number registration information for the purpose of spying on called numbers associated with various participating parties. The likelihood of this occurring on a large scale is small, because it might require a prohibitive (and easily-detectable) number of VIPR servers to capture all of the number registrations of a region under surveillance; however, more targeted attacks are feasible and should be recognized as a potential security consideration.

This security breach can occur because all registrations are considered equally untrusted, and they will be verified by establishing a TCP connection between the VIPR server of the source of the call and the VIPR server that stored the registration for a particular phone number. Multiple pieces of identifying information are necessarily leaked in this verification process, but it is specifically easy to identify the enterprise originating the TCP connection by comparing its source address to public registry data (such as in-addr.arpa).

For destination phone numbers using VIPR, the vulnerability arises because the RELOAD overlay permits multiple entities to register for the same number. The VIPR server at the source of the call may therefore discover multiple candidate registrations; although malicious servers registering themselves will not possess the call details necessary to generate a shared secret, they may learn sensitive information merely through participating in the verification process. While it is possible that the real owner of the number may be tried first and prevent other registrations to be tried if successful, an attacker could register from multiple VIPR servers in order to improve their chances of receiving a verification request. One could easily imagine an attacker determined to learn who will call a particular number generating a large set of registrations that would make it very unlikely for the authentic server to be selected first; with enough such registrations it might effectively become a denial of service attack. Note however that this problem is limited to server discovery: as soon as the real owner sends a SIP route and ticket back, the malicious VIPR server would no longer receive any information about the calls between the enterprise and the destination number, with exception of the periodic renewal of the ticket.

The possible disclosed information includes more than the just the connection verification. Here is a list of potential leaked information:

If the malicious VIPR servers leverage a different VServiceId for each registered phone number, the called number is always leaked.

The called number is leaked during the validation process for both methods A and B [draft-petithuguenin-vipr-pvp-04, Section 7.2.1, Section 7.2.2].

For method A, the caller-ID is leaked (this is encrypted, but it is possible to decrypt).

For method B, a random time in the middle of the call is leaked.

For method C, the rounded start and stop time of the call are leaked.

The source IP address of the TCP connection for the PVP transaction is always leaked.

The addr_port in the AppAttachReq RELOAD message that was used to establish the TCP connection is leaked. [draft-ietf-p2psip-base-24]

The certificate of the signer of the AppAttachReq RELOAD message is leaked. While the certificate does not contain information about the sender, but it always contains the Node-ID, which can always be resolved to an IP address by using an Attach request.

8. IANA Considerations

This specification does not require any actions from IANA.

9. Acknowledgements

Thanks for review comments from Ken Fischer, Rob Maidhof, Michael Procter, Eric Burger, Richard Barnes and others. Thanks to Theo Zourzouvillys for pointing out the 5th theft of phone numbers attack as described in Section 7.2 .

10. References

10.1. Normative References

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

10.2. Informative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[I-D.ietf-p2psip-base]
Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-26 (work in progress), February 2013.

[I-D.petithuguenin-vipr-reload-usage]
Petit-Huguenin, M., Rosenberg, J., and C. Jennings, "A Usage of Resource Location and Discovery (RELOAD) for Public Switched Telephone Network (PSTN) Verification", draft-petithuguenin-vipr-reload-usage-04 (work in progress), March 2012.

[I-D.petithuguenin-vipr-framework]
Petit-Huguenin, M., Jennings, C., and J. Rosenberg, "Verification Involving PSTN Reachability (VIPR): Framework", draft-petithuguenin-vipr-framework-00 (work in progress), October 2011.

[I-D.petithuguenin-vipr-sip-antispam]
Petit-Huguenin, M., Rosenberg, J., and C. Jennings, "Session Initiation Protocol (SIP) Extensions for Blocking VoIP Spam Using PSTN Validation", draft-petithuguenin-vipr-sip-antispam-03 (work in progress), January 2012.

[I-D.jennings-vipr-vap]
Jennings, C., Rosenberg, J., and M. Petit-Huguenin, "Verification Involving PSTN Reachability: The ViPR Access Protocol (VAP)", draft-jennings-vipr-vap-02 (work in progress), March 2012.

[I-D.petithuguenin-vipr-pvp]
Petit-Huguenin, M., Rosenberg, J., and C. Jennings, "The Public Switched Telephone Network (PSTN) Validation Protocol (PVP)", draft-petithuguenin-vipr-pvp-04 (work in progress), March 2012.

[I-D.petithuguenin-vipr-proportional-quota]

Petit-Huguenin, M., Rosenberg, J., and C. Jennings, "Proportional Quota in REsource LOcation And Discovery (RELOAD)", draft-petithuguenin-vipr-proportional-quota-00 (work in progress), October 2011.

[RFC2543] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[E.164] ITU-T, "The International Public Telecommunication Number Plan", Recommendation E.164, May 1997.

[RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", RFC 5039, January 2008.

[RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, March 2011.

[RFC5067] Lind, S. and P. Pfautz, "Infrastructure ENUM Requirements", RFC 5067, November 2007.

[RFC5054] Taylor, D., Wu, T., Mavrogiannopoulos, N., and T. Perrin, "Using the Secure Remote Password (SRP) Protocol for TLS Authentication", RFC 5054, November 2007.

[RFC6873] Salgueiro, G., Gurbani, V., and A. Roach, "Format for the Session Initiation Protocol (SIP) Common Log Format (CLF)", RFC 6873, February 2013.

[I-D.jones-insipid-session-id]
Jones, P., Pearce, C., Polk, J., and G. Salgueiro, "End-to-End Session Identification in IP-Based Multimedia Communication Networks", draft-jones-insipid-session-id-02 (work in progress), February 2013.

[I-D.ietf-insipid-session-id-reqts]
Jones, P., Salgueiro, G., Polk, J., Liess, L., and H. Kaplan, "Requirements for an End-to-End Session Identification in IP-Based Multimedia Communication Networks", draft-ietf-insipid-session-id-reqts-08 (work in progress), July 2013.

Appendix A. Changes since last version

This section must be removed before publication as an RFC.

Modifications between jennings-04 and jennings-03:

1. Updating references to SIPCLF and Session ID (INSIPID) documents.

Modifications between jennings-03 and jennings-02:

1. Reworded REQ -11 to clarify that in the case of call failures (i.e., IP calls), the system should fallback to inter-domain calling prior to VIPR.

2. Deleted REQ-12 (Handover) since it's really not specific functionality provided by VIPR.

3. Moved some text from the -01 version in the Technical Overview section back into the doc (not sure why it was removed previously).

4. Other editorial changes:

   - Added a Terminology section.

   - Clarified the use of the term "Call Agent".

   - Reworded discussion of email in section 2.2 (i.e., it's not useless).

   - Either changed or removed altogether terms like "neat", "clever", "incredible", "enormous" and any text that read like marketing literature as much as possible.

   - Removed some of the more subjective and superfluous language - i.e., condensed the text to be more concise (Section 5.2 and many others per the previous change)

   - Deleted explicit reference to "SIP Trunking" as the statement didn't introduce additional information in that paragraph and the term is not defined in this document.

   - and other minor editorial fixes.

Modifications between jennings-02 and jennings-01:

1. Sections 6,7,8 moved to new VIPR framework document.

2. Editorial changes.

3. Clarifications to re-enforce that the primary objective is not PSTN bypass but rather to enable enhanced services such as video between domains. Changed "VoIP" to "SIP" since the focus is not specifically voice.

4. Added reference for new framework document.

5. Section 5.3: Added references to other documents as appropriate - e.g., -pvp, -spam, etc.

6. Moved validation diagrams and text (from 5.3.4) into Validation and caching section (5.3.3).

7. Condensed discussion of spam in section 5.3.3 and updated SPAM section in security section.

Modifications between jennings-01 and rosenberg-04:

- Not specified.

Modifications between rosenberg-04 and rosenberg-03

- Nits.

- Shorter I-Ds references.

- Changed phone numbers to follow E.123 presentation.

- Expanded P2P initialisms.

- Uses +1 408 555 prefix for phone numbers in examples.

Authors' Addresses

Mary Barnes
Polycom
TX
US

Email: mary.ietf.barnes@gmail.com

Cullen Jennings
Cisco
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

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

Jonathan Rosenberg
jdrosen.net
Monmouth, NJ
US

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

Marc Petit-Huguenin
Unaffiliated

Email: petithug@acm.org