SIPPING
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2008

H. Tschofenig
Nokia Siemens Networks
D. Wing
Cisco
H. Schulzrinne
Columbia University
T. Froment
Alcatel-Lucent
G. Dawirs
University of Namur

July 8, 2007

A Document Format for Expressing Authorization Policies to tackle Spam and Unwanted Communication for Internet Telephony
draft-tschofenig-sipping-spit-policy-01.txt

Status of this Memo

Copyright Notice

Abstract

SPAM, defined as sending unsolicited messages to someone in bulk, might be a problem on SIP open-wide deployed networks. The responsibility for filtering or blocking calls can belong to different elements in the call flow and may depend on various factors. This document defines an authorization based policy language that allows end users to upload anti-SPIT policies to intermediaries, such as SIP proxies. These policies mitigate unwanted SIP communications. It extends the Common Policy authorization framework with additional conditions and actions. The new conditions match a particular Session Initiation Protocol (SIP) communication pattern based on a number of attributes. The range of attributes includes information provided, for example, by SIP itself, by the SIP identity mechanism, by information carried within SAML assertions.

Table of Contents

1. Introduction

The problem of SPAM for Internet Telephony (SPIT) is an imminent challenge and only the combination of several techniques can provide a framework for dealing with unwanted communication, as stated in [I-D.jennings-sip-hashcash].

One important building block is to have a mechanism that can instruct SIP intermediaries to react differently on incoming requests based on policies. Different entities, such as end users, parents on behalf of their children, system administrators in enterprise networks, etc., might create and modify authorization policies. The conditions in these policies can be created from many sources but some information elements are more important than others. For example, there is reason to believe that applying authorization policies based on the authenticated identity is an effective way to accept a communication attempt to deal with unsolicited communication. Authentication based on the SIP identity mechanism, see [RFC4474], is one important concept.

The requirements for the authorization policies described in this document are outlined in [I-D.froment-sipping-spit-requirements]. A framework document is available at [I-D.tschofenig-sipping-framework-spit-reduction].

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 RFC 2119 [RFC2119].

This document reuses the terminology from RFC 4745 [RFC4745]:

Rule maker:

The RM is an entity that creates the authorization policies that react to unwanted connection attempts. The rule maker might be an end user that owns the device, a VoIP service provider, a person with a relationship to the end user (e.g., the parents of a child using a mobile phone). A standardized policy language is needed when the creation, modification and deletion of authorization policies are not only a local matter.

Authorization policy:

An authorization policy is given by a rule set. A rule set contains an unordered list of rules. Each rule has a condition, an action and a transformation component. The terms 'authorization policy', 'policy', 'rule set', 'authorization policy rule', 'policy rule' and 'rule' are used interchangeably. Authorization policies can be applied at the end host and/or by intermediaries.

Permission:

The term permission refers to the action and transformation components of a rule.

We use the term 'Recipient' for the entity that is target of the communication attempt of a sender.

3. Generic Processing

3.1. Structure of SPIT Authorization Documents

A SPIT authorization document is an XML document, formatted according to the schema defined in RFC 4745 [RFC4745]. SPIT authorization documents inherit the MIME type of common policy documents, application/auth-policy+xml. As described in [RFC4745], this document is composed of rules which contain three parts - conditions, actions, and transformations. Each action or transformation, which is also called a permission, has the property of being a positive grant to the authorization server to perform the resulting actions, be it allow, block etc . As a result, there is a well-defined mechanism for combining actions and transformations obtained from several sources. This mechanism therefore can be used to filter connection attempts thus leading to effective SPIT prevention.

3.2. Rule Transport

Policies are XML documents that are stored at a Proxy Server or a dedicated device. The Rule Maker therefore needs to use a protocol to create, modify and delete the authorization policies defined in this document. Such a protocol is available with the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [RFC4825].

4. Condition Elements

This section describes the additional enhancements of the conditions-part of the rule. This document inherits the Common Policy functionality, including <identity>, <validity>, and <sphere> conditions.

Note that, as discussed in [RFC4745], a permission document applies to a translation if all the expressions in its conditions part evaluate to TRUE.

4.1. Identity

Although the <identity> element is defined in [RFC4745], that specification indicates that the specific usages of the framework

document need to define details that are protocol and usage specific. In particular, it is necessary for a usage of the common policy framework to:

- Define acceptable means of authentication.
- Define the procedure for representing the identity as a URI or IRI [RFC3987].

This sub-section defines those details for systems based on [RFC3856].

4.1.1. Acceptable Forms of Authentication

When used with SIP, a request is considered authenticated if one of the following techniques is used:

SIP Digest:

> The proxy has authenticated the sender using SIP [RFC3261] digest authentication [RFC2617]. However, if the anonymous authentication described on page 194 of RFC 3261 [RFC3261] was used, the sender is not considered authenticated.

Asserted Identity:

> If a request contains a P-Asserted-ID header field [RFC3325] and the request is coming from a trusted element, the sender is considered authenticated.

Cryptographically Verified Identity:

> If a request contains an Identity header field as defined in [RFC4474], and it validates the From header field of the request, the request is considered to be authenticated. Note that this is true even if the request contained a From header field of the form sip:anonymous@example.com. As long as the signature verifies that the request legitimately came from this identity, it is considered

authenticated.

An anonymous From header field with RFC 4474 [RFC4474] is considered authenticated, while anonymous digest is not considered authenticated, because the former still involves the usage of an actual username and credential as part of an authentication operation in the originating domain.

4.1.2. Computing a URI for the Sender

For messages that are authenticated using SIP Digest, the identity of the sender is set equal to the address of record (AoR) for the user that has authenticated themselves. The AoR is always a URI, and can be either a SIP URI or tel URI [RFC3966]. For example, consider the following "user record" in a database:

```
SIP AOR: sip:alice@example.com
digest username: ali
digest password: f779ajvvh8a6s6
digest realm: example.com
```

If the proxy server receives an INVITE, challenges it with the realm set to "example.com", and the subsequent INVITE contains an Authorization header field with a username of "ali" and a digest response generated with the password "f779ajvvh8a6s6", the identity used in matching operations is "sip:alice@example.com".

For messages that are authenticated using RFC 3325 [RFC3325], the identity of the sender is equal to the URI in the P-Asserted-ID header field. If there are multiple values for the P-Asserted-ID header field (there can be one sip URI and one tel URI [RFC3966]), then each of them is used for the comparisons outlined in [RFC4745], and if either of them match a <one> or <except> element, it is considered a match.

For messages that are authenticated using the SIP Identity mechanism [RFC4474], identity of the sender is equal to the SIP URI in the From header field of the request, assuming that the signature in the Identity header field has been validated.

In SIP systems, it is possible for a user to have aliases - that is,

there are multiple SIP AoRs "assigned" to a single user. In terms of this specification, there is no relationship between those aliases. Each would look like a different user. This will be the consequence for systems where the sender is in a different domain than the recipient. However, even if the sender and recipient are in the same domain, and the proxy server knows that there are aliases for the sender, these aliases are not mapped to each other or used in any way.

SIP also allows for anonymous identities. If a message is anonymous because the digest challenge/response used the "anonymous" username, the message is considered unauthenticated and will match only an empty <identity> element. If a message is anonymous because it contains a Privacy header field [RFC3323], but still contains a P-Asserted-ID header field, the identity in the P-Asserted-ID header field is still used in the authorization computations; the fact that the message was anonymous has no impact on the identity processing. However, if the message had traversed a trust boundary and the P-Asserted-ID header field and the Privacy header field had been removed, the message will be considered unauthenticated when it arrives at the proxy server. Finally, if a message contained an Identity header field that was validated, and the From header field contained a URI of the form sip:anonymous@example.com, then the sender is considered authenticated, and it will have an identity equal to sip:anonymous@example.com. Had such an identity been placed into a <one> or <except> element, there will be a match.

It is important to note that SIP frequently uses both SIP URI and tel URI [RFC3966] as identifiers, and to make matters more confusing, a SIP URI can contain a phone number in its user part, in the same format used in a tel URI. The sender's identity that is a SIP URI with a phone number will not match the <one> and <except> conditions whose 'id' is a tel URI with the same number. The same is true in the reverse. If the sender's identity is a tel URI, this will not match a SIP URI in the <one> or <except> conditions whose user part is a phone number. URIs of different schemes are never equivalent.

4.2. Sphere

The <sphere> element is defined in [RFC4745]. However, each application making use of the common policy specification needs to determine how the policy server computes the value of the sphere to be used in the evaluation of the condition.

To compute the value of <sphere>, the proxy server interacts with a presence server who knows whether at least one of the published presence documents includes the <sphere> element [RFC4480] as part of

the person data component [RFC4479], and all of those containing the element have the same value for it, that is the value used for the sphere in policy policy processing. If, however, the <sphere> element was not available to the presence server (and hence not for the proxy server), or it was present but had inconsistent values, its value is considered undefined in terms of policy processing.

4.3. SPIT Handling

The <spit-handling> element is a way to react on the execution of certain SPIT handling mechanisms. For example, a rule might indicate that a CAPTCHA has to be sent to the sender and the sender subsequently has to return the result. Depending on the outcome of the robot test the rules might enforce different actions. This element provides such a condition capability.

The <spit-handling> condition evaluates to TRUE if any of its child elements evaluate to TRUE, i.e., the results of the individual child element are combined using a logical OR.

The <spit-handling> element MAY contain zero or more <challenge> elements. The <challenge> elements has an attribute 'result' that either contains "SUCCESS" or "FAILURE".

4.4. Media List

The <media-list> condition evaluates to TRUE if any of its child elements evaluate to TRUE, i.e., the results of the individual child element are combined using a logical OR.

The <media-list> element SHOULD include either
- an <all-media-except> element or;
- a list of one or more >media> elements selected from the list of possible media elements below.

List of possible media elements:
- The <message-session> media element indicating session based messaging as defined in [I-D.ietf-simple-message-sessions];
- The <pager-mode-message> media element indicating pager mode message requests as defined in [RFC3428];
- The <file-transfer> media element indicating file transfer as defined in [I-D.ietf-mmusic-file-transfer-mech];
- The <audio> media element indicating a streaming media type as defined in [RFC3840];
- The <video> media element indicating a streaming media type as defined in [RFC3840];

- Any elements from any other namespaces defining a media element.

The <all-media-except> element MAY include a list of one or more >media> elements selected from the list of possible >media> elements above.

The <audio>, <video> and <message-session> elements:
- MAY include the <full-duplex> element indicating that media can be exchanged in both directions simultaneously;
- MAY include the <half-duplex> element indicating that media can be exchanged in only one direction at a time.

4.5. Method List

The <method-list> element contains one or more child elements <method> in order to provide matching capabilities of any SIP method invoked by the user can be used to filter incoming messages.

The <method-list> condition element evaluates to TRUE if any of its child elements evaluate to TRUE, i.e., the results of the individual child element are combined using a logical OR.

4.6. MIME List

The <mime-list> element contains one or more child <mime> child elements

The <mime-list> condition element evaluates to TRUE if any of its child elements evaluate to TRUE, i.e., the results of the individual child element are combined using a logical OR.

4.7. Presence Status

This condition evaluates to TRUE when the called user's current presence activity status is equal to the value in the <presence-status> element. Otherwise the condition evaluates to FALSE.

4.8. Rule Deactivated

The <rule-deactivated> condition always evaluates to FALSE. This can be used to deactivate a rule, without loosing information. By removing this condition the rule can be activated again.

4.9. Time Period Condition

The <time-period> element allows to make decisions based on the time, date and timezone. The <time> element may contain the following attributes:

tzid:

RFC 2445 [RFC2445] Time Zone Identifier

tzurl:

RFC 2445 [RFC2445] Time Zone URL

dtstart:

Start of interval (RFC 2445 [RFC2445] DATE-TIME)

dtend:

End of interval (RFC 2445 [RFC2445] DATE-TIME)

duration:

Length of interval (RFC 2445 [RFC2445] DURATION)

freq:

Frequency of recurrence ("secondly", "minutely", "hourly", "daily", "weekly", "monthly", or "yearly")

interval:

How often the recurrence repeats

until:

Bound of recurrence (RFC 2445 [RFC2445] DATE-TIME)

count:

Number of occurrences of recurrence

bysecond:

List of seconds within a minute

byminute:

List of minutes within an hour

byhour:

List of hours of the day

byday: List of days of the week

bymonthday:

List of days of the month

byyearday:

List of days of the year

byweekno:

List of weeks of the year

bymonth:

List of months of the year

wkstv:

First day of the work week

bysetpos:

List of values within set of events specified

The <time-period> is based on the description in CPL [RFC3880] and furthemore based closely on the specification of recurring intervals of time in the Internet Calendaring and Scheduling Core Object Specification (iCalendar COS), RFC 2445 [RFC2445].

This allows policies to be generated automatically from calendar books. It also allows us to re-use the extensive existing work specifying time intervals.

If future standards-track documents are published that update or obsolete RFC 2445 [RFC2445], any changes or clarifications those documents make to recurrence handling apply to CPL time-switches as well.

An algorithm to determine whether an instant falls within a given recurrence is given in Appendix A of RFC 3880 [RFC3880].

The <time-period> element takes two optional attributes, "tzid" and

"tzurl", both of which are defined in Sections 4.8.3.1 and 4.8.3.5 of RFC 2445 [RFC2445]. The "tzid" is the identifying attribute by which a time zone definition is referenced. If it begins with a forward slash (solidus), it references a to-be-defined global time zone registry; otherwise it is locally-defined at the server. The "tzurl" attribute gives a network location from which an up-to-date VTIMEZONE definition for the timezone can be retrieved.

While the content of the "tzid" attribute does not begin with a forward slash are locally defined, it is RECOMMENDED that servers support at least the naming scheme used by the Olson Time Zone database [OTZ]. Examples of timezone databases that use the Olson scheme are the zoneinfo files on most Unix-like systems, and the standard Java TimeZone class.

Servers SHOULD resolve "tzid" and "tzurl" references to time zone definitions at the time the policy is uploaded. They MAY periodically refresh these resolutions to obtain the most up-to-date

definition of a time zone. If a "tzurl" becomes invalid, servers SHOULD remember the most recent valid data retrieved from the URL.

If a script is uploaded with a "tzid" and "tzurl" which the CPL server does not recognize or cannot resolve, it SHOULD diagnose and reject this at script upload time. If neither "tzid" nor "tzurl" are present, all non-UTC times within this time switch should be interpreted as being "floating" times, i.e., that they are specified in the local timezone of the CPL server.

> Because of daylight-savings-time changes over the course of a year, it is necessary to specify time switches in a given timezone. UTC offsets are not sufficient, or a time-of-day routing rule which held between 9 am and 5 pm in the eastern United States would start holding between 8 am and 4 pm at the end of October.

The developers of tools used creating policy documents should be careful to handle correctly the intervals when local time is discontinuous, at the beginning or end of daylight-savings time. Note especially that some times may occur more than once when clocks are set back. The algorithm in Appendix A of RFC 3880 [RFC3880] is believed to handle this correctly.

The <time> element specifies a list of periods. They have two required attributes a:

- "dtstart", which specifies the beginning of the first period of the list,

- and exactly one of "dtend" or "duration", which specify the ending time or the duration of the period, respectively.

The "dtstart" and "dtend" attributes are formatted as iCalendar COS DATE-TIME values, as specified in Section 4.3.5 of RFC 2445 [RFC2445]. Only floating or UTC times can be used with time zones. The "duration" attribute is given as an iCalendar COS DURATION parameter, as specified in section 4.3.6 of RFC 2445 [RFC2445]. Both the DATE-TIME and the DURATION syntaxes are subsets of the corresponding syntaxes from ISO 8601 [ISO8601].

For a recurring interval, the "duration" attribute MUST be small enough such that subsequent intervals do not overlap. For non-recurring intervals, durations of any positive length are permitted. Zero-length and negative-length durations are not allowed.

If no other parameters are specified, a <time> element indicates only a single period of time. More complicated sets of period intervals are constructed as recurrences. A recurrence is specified by including the "freq" attribute, which indicates the type of recurrence rule. Parameters other than "dtstart", "dtend", and "duration" SHOULD NOT be specified unless "freq" is present, though servers SHOULD accept rules with such parameters present, and ignore the other parameters.

The "freq" parameter takes one of the following values: "secondly", to specify repeating periods based on an interval of a second or more, "minutely", to specify repeating periods based on an interval of a minute or more, "hourly", to specify repeating periods based on an interval of an hour or more, "daily", to specify repeating periods based on an interval of a day or more, "weekly", to specify repeating periods based on an interval of a week or more, "monthly", to specify repeating periods based on an interval of a month or more, and "yearly", to specify repeating periods based on an interval of a year or more. These values are not case-sensitive.

The "interval" attribute contains a positive integer representing how often the recurrence rule repeats. The default value is "1", meaning every second for a "secondly" rule, every minute for a "minutely" rule, every hour for an "hourly" rule, every day for a "daily" rule, every week for a "weekly" rule, every month for a "monthly" rule, and every year for a "yearly" rule.

The "until" attribute defines an iCalendar COS DATE or DATE-TIME value which bounds the recurrence rule in an inclusive manner. If the value specified by "until" is synchronized with the specified recurrence, this date or date-time becomes the last instance of the recurrence. If specified as a date-time value, then it MUST be

specified in UTC time format. If not present, and the "count" parameter is not also present, the recurrence is considered to repeat forever.

The "count" attribute defines the number of occurrences at which to range-bound the recurrence. The "dtstart" attribute counts as the first occurrence. The "until" and "count" attribute MUST NOT occur in the same <time> element.

The "bysecond" attribute specifies a comma-separated list of seconds within a minute. Valid values are 0 to 59. The "byminute" attribute specifies a comma-separated list of minutes within an hour. Valid values are 0 to 59. The "byhour" attribute specifies a comma-separated list of hours of the day. Valid values are 0 to 23.

The "byday" attribute specifies a comma-separated list of days of the week. "MO" indicates Monday, "TU" indicates Tuesday, "WE" indicates Wednesday, "TH" indicates Thursday, "FR" indicates Friday, "SA" indicates Saturday, and "SU" indicates Sunday. These values are not case-sensitive.

Each "byday" value can also be preceded by a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific day within the "monthly" or "yearly" recurrence. For example, within a "monthly" rule, +1MO (or simply 1MO) represents the first Monday within the month, whereas -1MO represents the last Monday of the month. If an integer modifier is not present, it means all days of this type within the specified frequency. For example, within a "monthly" rule, MO represents all Mondays within the month.

The "bymonthday" attribute specifies a comma-separated list of days of the month. Valid values are 1 to 31 or -31 to -1. For example, -10 represents the tenth to the last day of the month.

The "byyearday" attribute specifies a comma-separated list of days of the year. Valid values are 1 to 366 or -366 to -1. For example, -1 represents the last day of the year (December 31st) and -306 represents the 306th to the last day of the year (March 1st).

The "byweekno" attribute specifies a comma-separated list of ordinals specifying weeks of the year. Valid values are 1 to 53 or -53 to -1. This corresponds to weeks according to week numbering as defined in ISO 8601 [ISO8601]. A week is defined as a seven day period, starting on the day of the week defined to be the week start (see "wkst"). Week number one of the calendar year is the first week which contains at least four (4) days in that calendar year. This parameter is only valid for "yearly" rules. For example, 3 represents the third week of the year.

Note: Assuming a Monday week start, week 53 can only occur when January 1 is a Thursday or, for leap years, if January 1 is a Wednesday.

The "bymonth" attribute specifies a comma-separated list of months of the year. Valid values are 1 to 12.

The "wkst" attribute specifies the day on which the work week starts. Valid values are "MO", "TU", "WE", "TH", "FR", "SA" and "SU". This is significant when a "weekly" recurrence has an interval greater than 1, and a "byday" parameter is specified. This is also significant in a "yearly" recurrence when a "byweekno" parameter is specified. The default value is "MO", following ISO 8601 [ISO8601].

The "bysetpos" attribute specifies a comma-separated list of values which corresponds to the nth occurrence within the set of events specified by the rule. Valid values are 1 to 366 or -366 to -1. It MUST only be used in conjunction with another byxxx attribute. For example, "the last work day of the month" could be represented as:

```
<time dtstart="19970105T083000"
      freq="monthly"
      byday="MO,TU,WE,TH,FR"
      bysetpos="-1"
/>
```

Each "bysetpos" value can include a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific occurrence within the set of events specified by the rule.

If byxxx attribute values are found which are beyond the available scope (i.e., bymonthday="30" in February), they are simply ignored.

Byxxx attribute modify the recurrence in some manner. Byxxx rule parts for a period of time which is the same or greater than the frequency generally reduce or limit the number of occurrences of the recurrence generated. For example, freq="daily" bymonth="1" reduces the number of recurrence instances from all days (if the "bymonth" attribute is not present) to all days in January. Byxxx attribute for a period of time less than the frequency generally increase or expand the number of occurrences of the recurrence. For example, freq="yearly" bymonth="1,2" increases the number of days within the yearly recurrence set from 1 (if "bymonth" parameter is not present) to 2.

If multiple Byxxx attribute are specified, then after evaluating the

specified "freq" and "interval" attribute, the Byxxx attribute are

applied to the current set of evaluated occurrences in the following order: "bymonth", "byweekno", "byyearday", "bymonthday", "byday", "byhour", "byminute", "bysecond", and "bysetpos"; then "count" and "until" are evaluated.

Here is an example of evaluating multiple Byxxx attribute.

```
<time dtstart="19970105T083000"
      duration="10M"
      freq="yearly"
      interval="2"
      bymonth="1"
      byday="SU"
      byhour="8,9"
      byminute="30">
/>
```

First, the interval="2" would be applied to freq="yearly" to arrive at "every other year." Then, bymonth="1" would be applied to arrive at "every January, every other year." Then, byday="SU" would be applied to arrive at "every Sunday in January, every other year." Then, byhour="8,9" would be applied to arrive at "every Sunday in January at 8 AM and 9 AM, every other year." Then, byminute="30" would be applied to arrive at "every Sunday in January at 8:30 AM and 9:30 AM, every other year." Then the second is derived from "dtstart" to end up in "every Sunday in January from 8:30:00 AM to 8:40:00 AM, and from and 9:30:00 AM to 9:40:00 AM, every other year." Similarly, if the "byminute", "byhour", "byday", "bymonthday", or "bymonth" parameter were missing, the appropriate minute, hour, day, or month would have been retrieved from the "dtstart" parameter.

The iCalendar COS RDATE, EXRULE, and EXDATE recurrence rules are not specifically mapped to components of the <time-period> element. Equivalent functionality to the exception rules can be attained by using the ordering of rules to exclude times using earlier rules.

Section 4.4.1 of [RFC3880] provides some background of the differences to iCalendar and implementation issues.

5. Actions

As stated in [RFC4474], conditions are the 'if'-part of rules, whereas actions and transformations form their 'then'-part. The actions and transformations parts of a rule determine which operations the proxy server MUST execute on receiving a connection request attempt that matches all conditions of this rule. Actions

and transformations permit certain operations to be executed.

5.1. Execute Action

The <handling> element allows a couple of actions to be triggered, namely

Block Action:

> The block action states that this specific connection request MUST NOT be forwarded and a "403" forbidden message MUST be sent to the sender of the message.

Allow Action:

> The Allow action states that this specific connection request MUST be forwarded.

Furthermore, a couple of further mechanisms, such as computational puzzles mechanism (described in [I-D.jennings-sip-hashcash]), the consent framework (described in [I-D.ietf-sip-consent-framework]) etc. can be executed. Each mechanism needs to register a URI and the value of URI is placed in this field.

[Editior's Note: For editorial purposes the schema currently lists a few examples but in a non-URI format. When solution documents define these URIs then they can be used with this document.]

5.2. Forward To

The action supported in this section is forwarding of calls with the <forward-to> element that contains the following child elements:

target:

Specifies the address of the forwarding rule. It should be a valid SIP URI (RFC 3261 [RFC3261]) or TEL URI (RFC 3966 [RFC3966]).

6. Examples

This section provides a few examples for policy rules defined in this document. The example policy shows three rules with the rule id r1 - r4.

Rule r1 matches for authenticated identities from the domain "example.com", "example.org" and the single identity "sip:bob@good.example.net". For these conditions SIP messages are forwarded to the SIP UA as indicated with the <handling> element.

Rule r2 indicates that for SIP messages where the identity has not been verifiable the hash cash mechanism [I-D.jennings-sip-hashcash] and CAPTCHAs [I-D.tschofenig-sipping-captcha] are applied (see the 'hashcash' and the 'captcha' token in the <execute> element).
Rule r3 contains the <spit-handling> element with the <challenge> child element. This rule evaluates to TRUE if the sender returned a valid hash cash or a valid CAPTCHA result. The action part of the rule indicates that the call is then forwarded to the answering machine, namely sip:answering-machine@home.foo-bar.com.
Rule r4 blocks the call if sender provided a wrong hash cash or CAPTCHA result.

Rule r1 and r2 are valid only from 2007-01-01T01:00:00+01:00 to 2007-07-01T24:00:00+01:00.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
  xmlns:spit="urn:ietf:params:xml:ns:spit-policy"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<rule id="r1">
  <conditions>
    <identity>
      <one id="sip:bob@good.example.net"/>
      <many domain="example.com"/>
      <many domain="example.org"/>
    </identity>
    <validity>
      <from>2007-01-01T01:00:00+01:00</from>
      <until>2007-07-01T24:00:00+01:00</until>
    </validity>
  </conditions>
  <actions>
    <spit:execute>allow</spit:execute>
  </actions>
  <transformations/>
</rule>

<rule id="r2">
  <conditions>
    <validity>
```



```
      <from>2007-01-01T01:00:00+01:00</from>
      <until>2007-07-01T24:00:00+01:00</until>
    </validity>
  </conditions>
  <actions>
    <spit:execute>hashcash</spit:execute>
    <spit:execute>captcha</spit:execute>
  </actions>
  <transformations/>
</rule>

<rule id="r3">
  <conditions>
    <spit:spit-handling>
      <challenge result="SUCCESS">hashcash</challenge>
      <challenge result="SUCCESS">captcha</challenge>
    </spit:spit-handling>
  </conditions>
  <actions>
```

```
        <spit:forward-to>
          <target>sip:answering-machine@home.foo-bar.com
          </target>
        </spit:forward-to>
      </actions>
      <transformations/>
    </rule>

    <rule id="r4">
      <conditions>
        <spit:spit-handling>
            <challenge result="FAILURE">hashcash</challenge>
            <challenge result="FAILURE">captcha</challenge>
        </spit:spit-handling>
      </conditions>
      <actions>
        <spit:execute>block</spit:execute>
      </actions>
      <transformations/>
    </rule>

  </ruleset>
```

7. XML Schema

This section contains the XML schema that defines the policies schema described in this document. This schema extends the Common Policy schema (see [RFC4474]) by introducing new members of the <condition>

and <action> elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:spit-policy"
  xmlns:spit="urn:ietf:params:xml:ns:spit-policy"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- This import brings in the XML language attribute xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
```

```
      schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <xs:import namespace="urn:ietf:params:xml:ns:common-policy"/>

  <!-- Conditions -->

  <xs:element name="method-list">
    <xs:complexType>
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:sequence>
            <xs:element name="method" type="spit:method-type"
              minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax"
              minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="spit-handling">
    <xs:complexType>
          <xs:sequence>
            <xs:element name="challenge" type="spit:challenge-type"
              minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax"
              minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="result" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="SUCCESS"/>
                <xs:enumeration value="FAILURE"/>
              </xs:restriction>
            </xs:simpleType>
```



```
          </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="mime-list">
```

```
      <xs:complexType>
        <xs:complexContent>
          <xs:restriction base="xs:anyType">
            <xs:sequence>
              <xs:element name="mime" type="spit:mime-type"
                minOccurs="0" maxOccurs="unbounded"/>
              <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>

    <xs:element name="media-list">
      <xs:complexType>
        <xs:complexContent>
          <xs:restriction base="xs:anyType">
            <xs:sequence>
              <xs:element name="media" type="spit:media-type"
                minOccurs="0" maxOccurs="unbounded"/>
              <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>

    <xs:element name="rule-deactivated"
      type="spit:empty-element-type"/>

    <xs:complexType name="empty-element-type"/>

    <xs:element name="presence-status"
      type="spit:presence-status-activity-type"/>

    <xs:simpleType name="presence-status-activity-type">
      <xs:restriction base="xs:string"/>
    </xs:simpleType>

    <xs:simpleType name="media-type">
      <xs:restriction base="xs:string"/>
```

```
</xs:simpleType>

<xs:simpleType name="challenge-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:simpleType name="mime-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:simpleType name="method-type">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:element name="time-period" type="spit:TimeSwitchType"/>

<xs:simpleType name="YearDayType">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="-366"/>
        <xs:maxInclusive value="-1"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
        <xs:maxExclusive value="366"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>

</xs:simpleType>
<xs:simpleType name="DayType">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="[m|M][o|O]"/>
    <xs:pattern value="[t|T][u|U]"/>
    <xs:pattern value="[w|W][e|E]"/>
    <xs:pattern value="[t|T][h|H]"/>
    <xs:pattern value="[f|F][r|R]"/>
    <xs:pattern value="[s|S][a|A]"/>
    <xs:pattern value="[s|S][u|U]"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="TimeType">
  <xs:annotation>
    <xs:documentation>Exactly one of the two attributes
      "dtend" and "duration" must occur. None of
```

```
          the attributes following freq are meaningful
          unless freq appears. </xs:documentation>
      </xs:annotation>
      <xs:attribute name="dtstart" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation>RFC 2445 DATE-TIME</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="dtend" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>RFC 2445 DATE-TIME</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="duration" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>RFC 2445 DURATION</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="freq" type="spit:FreqType" use="optional"/>
      <xs:attribute name="interval"
        type="xs:positiveInteger" default="1"/>
      <xs:attribute name="until" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>RFC 2445 DATE-TIME</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="count"
        type="xs:positiveInteger" use="optional"/>
      <xs:attribute name="bysecond"
        type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>Comma-separated list of
            seconds within a minute. Valid values are 0 to
          59.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="byminute"
        type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation>Comma-separated list of
            minutes within an hour. Valid values are 0 to
          59.</xs:documentation>
```

```
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="byhour" type="xs:string" use="optional">
      <xs:annotation>
        <xs:documentation>Comma-separated list of
          hours of the day. Valid values are 0 to
```



```
          23.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="byday" type="xs:string" use="optional">
      <xs:annotation>
        <xs:documentation>Comma-separated list of days of the week.
          Valid values are "MO", "TU",
          "WE", "TH", "FR", "SA" and "SU". These values are
          not case-sensitive. Each can be preceded
          by a positive (+n) or negative (-n) integer.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="bymonthday" type="xs:string" use="optional">
      <xs:annotation>
        <xs:documentation>Comma-separated list of days of the month.
          Valid values are 1 to 31 or -31
          to -1.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="byyearday" type="xs:string" use="optional">
      <xs:annotation>
        <xs:documentation>Comma-separated list of days of the year.
          Valid values are 1 to 366 or
          -366 to -1.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="byweekno" type="xs:string" use="optional">
      <xs:annotation>
        <xs:documentation>Comma-separated list of ordinals
          specifying weeks of the year. Valid
          values are 1 to 53 or -53 to -1.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="bymonth" type="xs:string" use="optional">
```

```
        <xs:annotation>
          <xs:documentation>Comma-separated list of months of the year.
            Valid values are 1 to
          12.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="wkst" type="spit:DayType" default="MO"/>
      <xs:attribute name="bysetpos" type="spit:YearDayType"/>
      <xs:anyAttribute namespace="##any" processContents="lax"/>

    </xs:complexType>
    <xs:simpleType name="TZIDType">
      <xs:restriction base="xs:string"/>
```



```
    </xs:simpleType>
    <xs:simpleType name="TZURLType">
      <xs:restriction base="xs:anyURI"/>
    </xs:simpleType>

    <xs:complexType name="TimeSwitchType">
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:sequence>
            <xs:element name="time" type="spit:TimeType"
              minOccurs="1" maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="tzid" type="spit:TZIDType"/>
          <xs:attribute name="tzurl" type="spit:TZURLType"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>

    <xs:simpleType name="FreqType">
      <xs:restriction base="xs:NMTOKEN">
        <xs:pattern value="[s|S][e|E][c|C][o|O][n|N][d|D][l|L][y|Y]"/>
        <xs:pattern value="[m|M][i|I][n|N][u|U][t|T][e|E][l|L][y|Y]"/>
        <xs:pattern value="[h|H][o|O][u|U][r|R][l|L][y|Y]"/>
        <xs:pattern value="[d|D][a|A][i|I][l|L][y|Y]"/>
        <xs:pattern value="[w|W][e|E][e|E][k|K][l|L][y|Y]"/>
        <xs:pattern value="[m|M][o|N][n|N][t|T][h|H][l|L][y|Y]"/>
        <xs:pattern value="[y|Y][e|E][a|A][r|R][l|L][y|Y]"/>
      </xs:restriction>
```

```
    </xs:simpleType>

    <!-- Action -->

        <xs:element name="execute">
          <xs:simpleType>
            <xs:restriction base="xs:string">
            </xs:restriction>
          </xs:simpleType>
        </xs:element>

    <xs:element name="forward-to" type="spit:forward-to-type"/>

    <xs:complexType name="forward-to-type">
      <xs:sequence>
        <xs:element name="target" type="spit:target-type"/>
        <xs:any namespace="##other" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
```



```
    <xs:simpleType name="target-type">
      <xs:restriction base="xs:anyURI"/>
    </xs:simpleType>

  </xs:schema>
```

8. XCAP USAGE

The following section defines the details necessary for clients to manipulate SPIT authorization documents from a server using XCAP.

8.1. Application Unique ID

XCAP requires application usages to define a unique application usage ID (AUID) in either the IETF tree or a vendor tree. This specification defines the "Spit-policy" AUID within the IETF tree, via the IANA registration in Section 9.

8.2. XML Schema

XCAP requires application usages to define a schema for their documents. The schema for Anti-SPIT authorization documents is described in Section 7.

8.3. Default Namespace

XCAP requires application usages to define the default namespace for their documents. The default namespace is urn:ietf:params:xml:ns:spit-policy.

8.4. MIME Type

XCAP requires application usages to defined the MIME type for documents they carry. Anti-SPIT privacy authorization documents inherit the MIME type of Common Policy documents, application/ auth-policy+xml.

8.5. Validation Constraints

This specification does not define additional constraints.

8.6. Data Semantics

This document discusses the semantics of Anti-SPIT authorization.

8.7. Naming Conventions

When a SIP Proxy receives a SIP message to route it towards to a specific user foo, it will look for all documents within http://[xcaproot]/spit-policy/users/foo, and use all documents found beneath that point to guide authorization policy.

8.8. Resource Interdependencies

This application usage does not define additional resource interdependencies.

8.9. Authorization Policies

This application usage does not modify the default XCAP authorization policy, which is that only a user can read, write or modify his/her own documents. A server can allow privileged users to modify documents that they do not own, but the establishment and indication of such policies is outside the scope of this document.

9. IANA Considerations

There are several IANA considerations associated with this specification.

9.1. Anti-SPIT Policy XML Schema Registration

URI: urn:ietf:params:xml:schema:spit-policy
Registrant Contact: Hannes Tschofenig (hannes.tschofenig@siemens.com).
XML: The XML schema to be registered is contained in Section 7. Its first line is

<?xml version="1.0" encoding="UTF-8"?>

and its last line is

</xs:schema>

9.2. Anti-SPIT Policy Namespace Registration

URI: urn:ietf:params:xml:ns:spit-policy
Registrant Contact: Hannes Tschofenig (hannes.tschofenig@siemens.com).

XML:

9.3. XCAP Application Usage ID

This section registers an XCAP Application Usage ID (AUID) according to the IANA procedures defined in [RFC4825].

Name of the AUID: spit-policy

Description: The rules defined in this documents describe ways to react on unwanted and unsolicted communication (including Spam).

10. Security Considerations

This document aims to make it simple for users to influence the behavior of SIP message routing with an emphasis on SPIT prevention. This document proposes a strawman proposal for conditions and actions that might be useful when it comes to allowing a UA to tell its proxies which messages it wants to receive and what tasks it wants those proxies to perform before sending a SIP request to the UA.

A couple of requirements are described in [I-D.froment-sipping-spit-requirements] and a general discussion about the available solution mechanisms is available with [I-D.ietf-sipping-spam]. This document offers the ability to glue the different solution pieces together.

Since this document uses the Common Policy framework it also inherits its capabilities, including the combining permission algorithm that is applied when multiple rules fire. Unauthorized access to the user's Anti-SPIT rules must be prevented to avoid the introduction of security vulnerabilities.

11. Contributors

We would like to thank Mayutan Arumaithurai (mayutan.arumaithurai@gmail.com) for his work on this document.

12. Acknowledgments

We would like to thank

- Jonathan Rosenberg, David Schwartz and Dan York for sharing their thoughts with us before the first version of this document was written.

- Miguel Garcia and Remi Denis-Courmont for their review comments to the -00 version.
- Mayutan Arumaithurai for his editing help with the -00 version.
- Poikselka Miikka, Isomaki Markus, Jari Mutikainen, Jean-Marie Stupka, and Antti Laurila for their comments and for pointing us to specifications outside the IETF.

This document intentionally re-uses concept from existing documents. In particular, we reused

- ideas from SIEVE [RFC3028], a mail filtering language.
- the text in Section 4.9 from the Call Processing Language (CPL) [RFC3880]. The difference between CPL and this document is that CPL has a more procedural approach, while this proposal is matching-based.
- text in Section 4.1 from [I-D.ietf-simple-presence-rules].
- content of Section 5.2, Section 4.8 and Section 4.7 is reused from [ETSI-TS-183-004].
- text of Section 4.4 from [OMA-TS-XDM_Shared_Policy]

13. References

13.1. Normative References

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

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[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.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[RFC3856] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, August 2004.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

[RFC4412] Schulzrinne, H. and J. Polk, "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, February 2006.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC4479] Rosenberg, J., "A Data Model for Presence", RFC 4479, July 2006.

[RFC4480] Schulzrinne, H., Gurbani, V., Kyzivat, P., and J. Rosenberg, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", RFC 4480, July 2006.

[RFC4745] Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., Polk, J., and J. Rosenberg, "Common Policy: A Document Format for Expressing Privacy Preferences", RFC 4745, February 2007.

[RFC4825] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", RFC 4825, May 2007.

13.2. Informative References

[ETSI-TS-183-004] ETSI, "TS 183 004, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); PSTN/ISDN simulation services: Communication Diversion (CDIV); Protocol specification", 2007.

[I-D.froment-sipping-spit-requirements]
Froment, T., "Requirements for Authorization Policies to tackle Spam for Internet Telephony and Unwanted Traffic", draft-froment-sipping-spit-requirements-00 (work in

progress), June 2007.

[I-D.ietf-ecrit-service-urn]
Schulzrinne, H., "A Uniform Resource Name (URN) for Services", draft-ietf-ecrit-service-urn-06 (work in progress), March 2007.

[I-D.ietf-mmusic-file-transfer-mech]
Garcia-Martin, M., "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer", draft-ietf-mmusic-file-transfer-mech-03 (work in progress), June 2007.

[I-D.ietf-simple-message-sessions]
Campbell, B., "The Message Session Relay Protocol", draft-ietf-simple-message-sessions-19 (work in progress), February 2007.

[I-D.ietf-simple-presence-rules]
Rosenberg, J., "Presence Authorization Rules", draft-ietf-simple-presence-rules-09 (work in progress), March 2007.

[I-D.ietf-sip-consent-framework]
Rosenberg, J., "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", draft-ietf-sip-consent-framework-02 (work in progress), July 2007.

[I-D.ietf-sipping-spam]
Jennings, C. and J. Rosenberg, "The Session Initiation Protocol (SIP) and Spam", draft-ietf-sipping-spam-04 (work in progress), February 2007.

[I-D.jennings-sip-hashcash]
Jennings, C., "Computational Puzzles for SPAM Reduction in

SIP", draft-jennings-sip-hashcash-05 (work in progress), June 2007.

[I-D.mahy-iptel-cpc]
Mahy, R., "The Calling Party's Category tel URI Parameter", draft-mahy-iptel-cpc-06 (work in progress), March 2007.

[I-D.rosenberg-sipping-service-identification]
Rosenberg, J., "Identification of Communications Services in the Session Initiation Protocol (SIP)", draft-rosenberg-sipping-service-identification-02 (work in

progress), May 2007.

[I-D.schubert-sipping-saml-cpc]
Schubert, S., "Conveying CPC using the SAML", draft-schubert-sipping-saml-cpc-02 (work in progress), July 2006.

[I-D.schwartz-sipping-spit-saml]
Schwartz, D., "SPAM for Internet Telephony (SPIT) Prevention using the Security Assertion Markup Language (SAML)", draft-schwartz-sipping-spit-saml-01 (work in progress), June 2006.

[I-D.tschofenig-sipping-captcha]
Tschofenig, H. and E. Leppanen, "Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) based Robot Challenges for the Session Initiation Protocol (SIP)", draft-tschofenig-sipping-captcha-00 (work in progress), July 2007.

[I-D.tschofenig-sipping-framework-spit-reduction]
Tschofenig, H., "A Framework for Reducing Spam for Internet Telephony", draft-tschofenig-sipping-framework-spit-reduction-00 (work in progress), June 2007.

[ISO8601] ISO (International Organization for Standardization), ""Data elements and interchange formats -- Information

interchange -- Representation of dates and times", ISO Standard ISO 8601:2000(E), International Organization for Standardization, Geneva, Switzerland,", December 2000.

[OMA-TS-XDM_Shared_Policy]
Open Mobile Alliance, "Shared Policy XDM Specification", 2007.

[OTZ] Eggert, P., "Sources for Time Zone and Daylight Saving Time Data, available at http://www.twinsun.com/tz/tz-link.htm", 2007.

[RFC2445] Dawson, F. and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998.

[RFC3028] Showalter, T., "Sieve: A Mail Filtering Language", RFC 3028, January 2001.

[RFC3266] Olson, S., Camarillo, G., and A. Roach, "Support for IPv6 in Session Description Protocol (SDP)", RFC 3266, June 2002.

[RFC3880] Lennox, J., Wu, X., and H. Schulzrinne, "Call Processing Language (CPL): A Language for User Control of Internet Telephony Services", RFC 3880, October 2004.

Authors' Addresses

Hannes Tschofenig
Nokia Siemens Networks
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@nsn.com
URI: http://www.tschofenig.com

Dan Wing

Cisco

Phone:
Email: dwing@cisco.com

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs@cs.columbia.edu
URI: http://www.cs.columbia.edu

Thomas Froment
Alcatel-Lucent
1, rue Ampere - BP 80056
Massy, Paris 91302
France

Email: Thomas.Froment@alcatel-lucent.fr

Geoffrey Dawirs
University of Namur
21, rue Grandgagnage
Namur B-5000
Belgique

Email: gdawirs@gdawirs.be

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).