                 The Session Initiation Protocol (SIP) and Spam
                        draft-ietf-sipping-spam-01

Status of this Memo

   By submitting this Internet-Draft, each author represents that any
   applicable patent or other IPR claims of which he or she is aware
   have been or will be disclosed, and any of which he or she becomes
   aware will be disclosed, in accordance with Section 6 of BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on January 18, 2006.

Abstract

   Spam, defined as the transmission of bulk unsolicited messages, has
   plagued Internet email. Unfortunately, spam is not limited to email.
   It can affect any system that enables user to user communications.
   The Session Initiation Protocol (SIP) defines a system for user to
   user multimedia communications. Therefore, it is susceptible to
   spam, just as email is. In this document, we analyze the problem of

   spam in SIP. We first identify the ways in which the problem is the
   same and the ways in which it is different from email. We then
   examine the various possible solutions that have been discussed for
   email and consider their applicability to SIP. Discussions on this
   draft should be directed at sipping@ietf.org.

Table of Contents

1. Introduction

Spam, defined as the transmission of bulk unsolicited email, has been
a plague on the Internet email system, rendering it nearly useless.
Many solutions have been documented and deployed to counter the
problem. None of these solutions is ideal. However, one thing is
clear: the spam problem would be much less significant had solutions
been deployed ubiquitously before the problem became widespread.

The Session Initiation Protocol (SIP) [2] is used for multimedia
communications between users, including voice, video, instant
messaging and presence. Although it has seen widespread deployment,
the deployments today have mostly been in disconnected islands.
Providers have not yet connected to each other in significant ways,
nor have they yet opened up access so as to allow receipt of SIP
messaging from the open Internet. Possibly as a result of this, SIP
networks have not yet been the target of any significant amount of
spam. However, we believe that it is just a matter of time.

It is important that the SIP community react now, rather than later,
and define and deploy anti-spam measures before the problem arises.
This document serves to help frame the problem of spam in SIP and
analyze the solution space in order to help determine a path forward.

2. Problem Definition

The spam problem in email is well understood, and we make no attempt
to further elaborate on it here. The question, however, is what is
the meaning of spam when applied to SIP? Since SIP covers a broad
range of functionality, there appear to be three related but
different manifestations:

Call Spam: This type of spam is defined as a bulk unsolicited set of
   session initiation attempts (i.e., INVITE requests), attempting to
   establish a voice, video, instant messaging [1] or other type of
   communications session. If the user should answer, the spammer
   proceeds to relay their message over the real time media. This is
   the classic telemarketer spam, applied to SIP.

IM Spam: This type of spam is similar to email. It is defined as a
   bulk unsolicited set of instant messages, whose content contains
   the message that the spammer is seeking to convey. IM spam is
   most naturally sent using the SIP MESSAGE [3] request. However,
   any other request which causes content to automatically appear on
   the user's display will also suffice. That might include INVITE
   requests with large Subject headers (since the Subject is
   sometimes rendered to the user), or INVITE requests with text or
   HTML bodies.

Presence Spam: This type of spam is similar to IM spam. It is
   defined as a bulk unsolicited set of presence requests (i.e.,
   SUBSCRIBE requests [4] for the presence event package [7]), in an
   attempt to get on the "buddy list" or "white list" of a user in
   order to send them IM or initiate other forms of communications.

There are many other SIP messages that a spammer might send.
However, most of the other ones do not result in content being
delivered to a user, nor do they seek input from a user. Rather,
they are answered by automata. OPTIONS is a good example of this.
There is little value for a spammer in sending an OPTIONS request,
since it is answered automatically by the UAS. No content is
delivered to the user, and they are not consulted.

In the sections below, we consider the likelihood of these various
forms of SIP spam. This is done in some cases by a rough cost
analysis. It should be noted that all of these analyses are
approximate, and serve only to give a rough sense of the order of
magnitude of the problem.

2.1 Call Spam

Will call spam occur? That is an important question to answer.
Clearly, it does occur in the existing telephone network, in the form
of telemarketer calls. Although these calls are annoying, they do
not arrive in the same kind of volume as email spam. The difference
is cost; it costs more for the spammer to make a phone call than it
does to send email. This cost manifests itself in terms of the cost
for systems which can perform telemarketer call, and in cost per
call.

Both of these costs are substantially reduced by SIP. A SIP call
spam application is easy to write. It is just a UAC that initiates,
in parallel, a large number of calls. If a call connects, the spam
application generates an ACK and proceeds to play out a recorded
announcement, and then it terminates the call. This kind of
application can be built entirely in software, using readily
available (and indeed, free) off the shelf components. It can run on
a low end PC and requires no special expertise to execute.

The cost per call is also substantially reduced. A normal
residential phone line allows only one call to be placed at a time.
If additional lines are required, a user must purchase more expensive
connectivity. Typically, a T1 or T3 would be required for a large
volume telemarketing service. That kind of access is very expensive
and well beyond the reach of an average user. A T1 line is
approximately US $250 per month, and about 1.5 cents per minute for
calls. T1 lines used only for outbound calls (such as in this case)

are even more expensive than inbound trunks due to the reciprocal
termination charges that a provider pays and receives.

There are two aspects to the capacity: the call attempt rate, and the
number of simultaneous successful calls that can be in progress. A
T1 would allow a spammer at most 24 simultaneous calls, and assuming
about 10s for each call attempt, about 2.4 call attempts per second.
At high volume calling, the per-minute rates far exceed the flat
monthly fee for the T1. The result is a cost of 250,000 microcents
for each successful spam delivery, assuming 10s of content.

With SIP, this cost is much reduced. Consider a spammer using a
typical broadband Internet connection that provides 500Kbps of
upstream bandwidth. Initiating a call requires just a single INVITE
message. Assuming, for simplicity's sake, that this is 1kB, a
500Kbps upstream DSL or cable modem connection will allow about 62
call attempts per second. A successful call requires enough
bandwidth to transmit a message to the receiver. Assuming a low
compression codec (say, G.723.1 at 5.6 Kbps), as many as 90
simultaneous calls can be in progress. With 10s of content per call,
that allows for 9 successful call attempts per second. This means
that a system could deliver a voice message successfully to users at
a rate of around 9 per second. If broadband access is around $50/
month, the cost per successful voice spam is about 215 microcents
each. This assumes that calls can be made 24 hours a day, which may
or may not be the case.

These figures indicate that SIP call spam is roughly three orders of
magntiude cheaper to send than traditional circuit-based telemarketer
calls. This low cost is certainly going to be very attractive to
spammers. Indeed, many spammers utilize computational and bandwidth
resources provided by others, by infecting their machines with
viruses that turn them into "zombies" that can be used to generate
spam. This can reduce the cost of call spam to nearly zero.

Even ignoring the zombie issue, this reduction in cost is even more
amplified for international calls. Currently, there is very little
telemarketing calls across international borders, largely due to the
large cost of making international calls. This is one of the reasons
why the "do not call list", a United States national list of numbers
that telemarketers cannot call - has been effective. The law only
affects U.S. companies, but since most telemarketing calls are
domestic, it has been effective. Unfortunately (and fortunately),
the IP network provides no boundaries of these sorts, and calls to
any SIP URL are possible from anywhere in the world. This will allow
for international spam at a significantly reduced cost.
International spam is likely to be even more annoying that national
spam, since it may arrive in languages that the recipient doesn't

even speak.

These figures assume that the primary limitation is the access
bandwidth and not CPU, disk, or termination costs. Termination costs
merit further discussion. Currently, most VoIP calls terminate on
the Public Switched Telephone Network (PSTN), and this termination
costs the originator of the call money. These costs are similar to
the per-minute rates of a T1. It ranges anywhere from half a cent to
three cents per minute, depending on volume and other factors.
However, equipment costs, training and other factors are much lower
for SIP-based termination than a T1, making the cost still lower than
circuit connectivity. Furthermore, the current trend in VoIP systems
is to make termination free for calls that never touch the PSTN, that
is, calls to actual SIP endpoints. Thus, as more and more SIP
endpoints come online (there are probably around 5 million
addressable SIP endpoints on the Internet as of writing), termination
costs will probably drop. Until then, SIP spam can be used in
concert with termination services for a lower cost form of
traditional telemarketer calls, made to normal PSTN endpoints.

This number (9 deliveries per second) is below the successful message
delivery rate of email [[NOTE: is there a figure for this]].
However, many spam messages are automatically deleted by filters or
users without ever being read. It is far more likely that a call
spam will be examined by a user if its delivered, due to the
difficulty in automated content filtering (see below). Thus, when
one examines the final figure of importance - the number of new
customers attracted per spam delivered, it is far from clear whether
call spam or email spam will be more effective.

Another part of the cost of spamming is collecting addresses.
Spammers have, over time, built up immense lists of email addresses,
each of the form user@domain, to which spam is directed. SIP uses
the same form of addressing, making it likely that email addresses
can easily be turned into valid SIP addresses. Telephone numbers
also represent valid SIP addresses, in that, in concert with a
termination provider, a spammer can direct SIP calls at traditional
PSTN devices. It is not clear whether email spammers have also been
collecting phone numbers as they perform their web sweeps, but it is
probably not hard to do so. Furthermore, unlike email addresses,
phone numbers are a finite address space and one that is fairly
densely packed. As a result, going sequentially through phone
numbers is likely to produce a fairly high hit rate. Thus, it seems
like the cost is relatively low for a spammer to obtain large numbers
of SIP addresses to which spam can be directed.

2.2 IM Spam

   IM spam is very much like email, in terms of the costs for deploying
   and generating spam. Assuming, for the sake of argument, a 1kB
   message to be sent and 500 Kbps of upstream bandwidth, thats 62
   messages per second. At $50/month, the result is 31 microcents per
   message. This is less than voice spam, but not substantially less.
   The cost is probably on par with email spam. However, IM is much
   more intrusive than email. In today's systems, IMs automatically pop
   up and present themselves to the user. Email, of course, must be
   deliberately selected and displayed. However, many IM systems employ
   white lists, which only allow spam to be delivered if the sender is
   on the white list. Thus, whether or not IM spam will be useful seems
   to depend a lot on the nature of the systems as the network is opened
   up. If they are ubiquitously deployed with white-list access, the
   value of IM spam is likely to be low.

   It is important to point out that there are two different types of IM
   systems. Page mode IM systems work much like email, with each IM
   being sent as a separate message. In session mode IM, there is
   signaling in advance of communication to establish a session, and
   then IMs are exchanged, perhaps point-to-point, as part of the
   session. The modality impacts the types of spam techniques that can
   be applied. Techniques for email can be applied identically to page
   mode IM, but session mode IM is more like telephony, and many
   techniques (such as content filtering) are harder to apply.

2.3 Presence Spam

   As defined above, presence spam is the generation of bulk unsolicited
   SUBSCRIBE messages. What would be the effect of such spam? Most
   presence systems provide some kind of consent framework. A watcher
   that has not been granted permission to see the user's presence will
   not gain access to their presence. However, the presence request is
   usually noted and conveyed to the user, allowing them to approve or
   deny the request. In SIP, this is done using the watcherinfo event
   package [8]. This package allows a user to learn the identity of the
   watcher, in order to make an authorization decision.

   Interestingly, this provides a vehicle for conveying information to a
   user. By generating SUBSCRIBE requests from identities such as
   sip:please-buy-my-product@spam.example.com, brief messages can be
   conveyed to the user, even though the sender does not have, and never
   will receive, permission to access presence. As such, presence spam
   can be viewed as a form of IM spam, where the amount of content to be
   conveyed is limited. The limit is equal to the amount of information
   generated by the watcher that gets conveyed to the user through the
   permission system.

This type of spam also shows up in consent frameworks used to prevent
call spam, as discussed in Section 3.4.

3. Solution Space

In this section, we consider the various solutions that might be
possible to deal with SIP spam. We primarily consider techniques
that have been employed to deal with email spam. It is important to
note that the solutions documented below are not meant to be an
exhaustive study of the spam solutions used for email but rather just
a representative set. We also consider some solutions that appear to
be SIP-specific.

3.1 Content Filtering

The most common form of spam protection used in email is based on
content filtering. These spam filters analyze the content of the
email, and look for clues that the email is spam. Bayesian spam
filters are in this category.

Unfortunately, this type of spam filtering is almost completely
useless for call spam. There are two reasons. First, in the case
where the user answers the call, the call is already established and
the user is paying attention before the content is delivered. The
spam cannot be analyzed before the user sees it. Second, if the
content is stored before the user accesses it (e.g., with voicemail),
the content will be in the form of recorded audio or video. Speech
and video recognition technology is not likely to be good enough to
analyze the content and determine whether or not it is spam. Indeed,
if a system tried to perform speech recognition on a recording in
order to perform such an analysis, it would be easy for the spammers
to make calls with background noises, poor grammar and varied
accents, all of which will throw off recognition systems. Video
recognition is even harder to do and remains primarily an area of
research.

Therefore, our conclusion is that the most successful form of anti-
spam measures used in email are almost useless for call spam.

IM spam, due to its similarity to email, can be countered with
content analysis tools. Indeed, the same tools and techniques used
for email will directly work for IM spam.

3.2 Black Lists

Black listing is an approach whereby the spam filter maintains a list
of addresses that identify spammers. These addresses include both
usernames (spammer@domain.com) and entire domains (spammers.com).

Pure blacklists are not very effective in email for two reasons.
First, email addresses are easy to spoof, making it easy for the
sender to pretend to be someone else. If the sender varies the
addresses they send from, the black list becomes almost completely
useless. The second problem is that, even if the sender doesn't
forge the from address, email addresses are in almost limitless
supply. Each domain contains an infinite supply of email addresses,
and new domains can be obtained for very low cost. Furthermore,
there will always be public providers that will allow users to obtain
identities for almost no cost (for example, Yahoo or AOL mail
accounts). The entire domain cannot be blacklisted because it
contains so many valid users. Blacklisting needs to be for
individual users. Those identities are easily changed.

As a result, as long as identities are easy to manufacture, black
lists will have limited effectiveness for email.

Blacklists are also likely to be ineffective for SIP spam.
Fortunately, SIP has much stronger mechanisms for inter-domain
authenticated identity than email has (see Section 4). Assuming
these mechanisms are used and enabled in inter-domain communications,
it becomes nearly impossible to forge sender addresses. However, it
still remains cheap to obtain a nearly infinite supply of addresses.

3.3 White Lists

White lists are the opposite of black lists. It is a list of valid
senders that a user is willing to accept email from. Unlike black
lists, a spammer can not change identities to get around the white
list. White lists are susceptible to address spoofing, but a strong
identity authentication mechanism can prevent that problem. As a
result, the combination of white lists and strong identity are a good
form of defense against spam.

However, they are not a complete solution, since they would prohibit
a user from ever being able to receive email from someone who was not
explicitly put on the white list. As a result, white lists require a
solution to the "introduction problem" - how to meet someone for the
first time, and decide whether they should be placed in the white
list. In addition to the introduction problem, white lists demand
time from the user to manage.

In IM systems, white lists have proven exceptionally useful at
preventing spam. This is due, in no small part, to the fact that the
white list exists naturally in the form of the buddy list. Users
don't have to manage this list just for the purposes of spam
prevention; it provides general utility, and assists in spam
prevention for free. IM systems also have strong identity mechanisms

   due to their closed nature. The introduction problem in these
   systems is solved with a consent framework, described below.

   The success of white lists in IM systems has applicability to SIP as
   well, more so than email. This is because SIP also provides a buddy
   list concept and has an advanced presence system as part of its
   specifications. Second, unlike email, but like IM systems, SIP can
   provide a much more secure form of authenticated identity, even for
   inter-domain communications. As a result, the problem of forged
   senders can be eliminated, making the white list solution feasible.

   The introduction problem remains, however. In email, techniques like
   the Turing tests have been employed for this purpose. Those are
   considered further in the sections below. As with email, a technique
   for solving the introduction problem would need to be applied in
   conjunction with a white list.

3.4 Consent-Based Communications

   A consent-based solution is used in conjunction with white or black
   lists. That is, if user A is not on user B's white or black list,
   and user A attempts to communicate with user B, user A's attempt is
   initially rejected, and they are told that consent is being
   requested. Next time user B connects, user B is informed that user A
   had attempted communications. User B can then authorize or reject
   user A.

   These kinds of consent-based systems are used widely in presence and
   IM but not in email. This is likely due to the need for a secure
   authenticated identity mechanism, which is a pre-requisite for this
   kind of solution. Since most of today's IM systems are closed,
   sender identities can be authenticated.

   This kind of consent-based communications has been standardized in
   SIP for presence, using the watcher information event package [8] and
   data format [9], which allow a user to find out that someone has
   subscribed. Then, the XML Configuration Access Protocol (XCAP) [11]
   is used, along with the XML format for presence authorization [12] to
   provide permission for the user to communicate. However, to date,
   these techniques have been applied strictly for presence.

   If they were extended to cover IM and calling, would it help? It is
   hard to say. At first glance, it would seem to help a lot. However,
   it might just change the nature of the spam. Instead of being
   bothered with content, in the form of call spam or IM spam, users are
   bothered with consent requests. A user's "communications inbox"
   might instead be filled with requests for communications from a
   multiplicity of users. Those requests for communications don't

convey much useful content to the user, but they can convey some. At
the very least, they will convey the identity of the requester. The
user part of the SIP URI allows for limited freeform text, and thus
could be used to convey brief messages. One can imagine receiving
consent requests with identities like,
"sip:please-buy-my-product-at-this-website@spam.example.com", for
example. Fortunately, traditional content-based filtering can be
applied to this type of information.

In order for the spammer to convey more extensive content to the
user, the user must explicitly accept the request, and only then can
the spammer convey the full content. This is unlike email spam,
where, even though much spam is automatically deleted, some
percentage of the content does get through, and is seen by users,
without their explicit consent that they want to see it. Thus, if
consent is required first, and nearly all users do not give consent
to spammers, the value in sending spam is reduced, and perhaps it
will cease.

As such, the real question is whether or not the consent system would
make it possible for a user to give consent to non-spammers and
reject spammers. Authenticated identity can help. A user in an
enterprise would know to give consent to senders in other enterprises
in the same industry, for example. However, in the consumer space,
if sip:bob@example.com tries to communicate with a user, how does
that user determine whether bob is a spammer or a long-lost friend
from high school? There is no way based on the identity alone. In
such a case, a useful technique is to grant permission for bob to
communicate but to make the permission is extremely limited. In
particular, bob may be granted permission to send no more than 200
words of text in a single IM, which he can use to identify himself,
so that the user can determine whether or not more permissions are
appropriate. However, this 200 words of text may be enough for a
spammer to convey their message, in much the same way they might
convey it in the user part of the SIP URI.

Thus, it seems that a consent-based framework, along with white lists
and black lists, cannot fully solve the problem for SIP, although it
does appear to help.

3.5 Reputation Systems

A reputation system is also used in conjunction with white or black
lists. Assume that user A is not on user B's white list, and they
attempt to contact user B. If a consent-based system is used, B is
prompted to consent to communications from A, a reputation score
might be displayed in order to help B decide whether or not they
should accept communications from A.

Traditionally, reputation systems are implemented in highly
centralized messaging architectures; the most widespread reputation
systems in messaging today have been deployed by monolithic instant
messaging providers (though many web sites with a high degree of
interactivity employ very similar concepts of reputation).
Reputation is calculated based on user feedback. For example, a
button on the user interface of the messaging client might empower
users to inform the system that a particular user is abusive. Of
course, the input of any single user has to be insufficient to ruin
one's reputation, but consistent negative feedback would give the
abusive user a negative reputation score.

Reputation systems have not enjoyed much success outside of the
instant messaging space. This is in part because few other
communications systems admit of the same degree of centralization and
monolithic control. That control, first of all, provides a
relatively strong identity assertion for users (since all users trust
a common provider, and the common provider is the arbiter of
authentication and identity). Secondly, it provides a single place
where reputation can be managed.

Reputation systems based on negative reputation scores suffer from
many of the same problems as black lists, since effectively the
consequence of having a negative reputation is that you are
blacklisted. If identities are very easy to acquire, a user with a
negative reputation will simply acquire a new one. Moreover,
negative reputation is generated by tattling, which requires users to
be annoyed enough to click the warning button. Additionally, it can
be abused. In some reputation systems, "reputation mafias"
consisting of large numbers of users routinely bully or extort
victims by threatening collectively to grant victims a negative
reputation.

Reputation systems based on positive reputation, where users praise
each other for being good, rather than tattling on each other for
being bad, have some similar drawbacks. Collectives of spammers, or
just one spammer who acquires a large number identities, could praise
one another in order to create an artifical positive reputation.
Users similarly have to overcome the inertia required to press the
"praise" button. Unlike negative reputation systems, however,
positive reputation is not circumvented when users require a new
identity, since basing authorization decisions on positive reputation
is essentially a form of whitelisting.

So, while positive reputation systems are superior to negative
reputation systems, they are far from perfect. Intriguingly, though,
combining presence-based systems with reputation systems leads to an
interesting fusion. The "buddy-list" concept of presence is, in

effect, a white list - and one can therefore probably infer that the
users on one's buddy list are people whom you are "praising". This
eliminates the problem of user inertia in the use of the "praise"
button, and automates the initial establishment of reputation.

And of course, your buddies in turn have buddies. Collectively, you
and your buddies (and their buddies, and so on) constitute a social
network of reputation. If there were a way to leverage this social
network, it would eliminate the need for centralization of the
reputation system. Your perception of a particular user's reputation
might be dependent on your relationship to them in the social
network: are they one buddy removed (strong reputation), four buddies
removed (weaker reputation), three buddies removed but connected to
you through several of your buddies, etc. This web of trust
furthermore would have the very desirable property that circles of
spammers adding one another to their own buddylists would not affect
your perception of their reputation unless their circle linked to
your own social network.

3.6 Address Obfuscation

Spammers build up their spam lists by gathering email addresses from
web sites and other public sources of information. One way to
prevent spam is to make your address difficult or impossible to
gather. Spam bots typically look for text in pages of the form
"user@domain", and assume that anything of that form is an email
address. To hide from such spam bots, many websites have recently
begun placing email addresses in an obfuscated form, usable to humans
but difficult for an automata to read as an email address. Examples
include forms such as, "user at example dot com" or "j d r o s e n a
t e x a m p l e d o t c o m".

These techniques are equally applicable to prevention of SIP spam,
and are likely to be as equally effective or ineffective in its
prevention.

It is worth mentioning that the source of addresses need not be a web
site - any publically accessible service containing addresses will
suffice. As a result, ENUM [10] has been cited as a potential gold
mine for spammers. It would allow a spammer to collect SIP and other
URIs by traversing the tree in e164.arpa and mining it for data.
This problem is mitigated in part if only number prefixes, as opposed
to actual numbers, appear in the DNS. Even in that case, however, it
provides a technique for a spammer to learn which phone numbers are
reachable through cheaper direct SIP connectivity.

3.7 Limited Use Addresses

A related technique to address obfuscation is limited use addresses.
In this technique, a user has a large number of email addresses at
their disposal. They give out different email addresses to different
people. Once spam begins arriving at an address, the user terminates
the address and replaces it with another.

This technique is equally applicable to SIP. One of the drawbacks of
the approach is that it can make it hard for people to reach you; if
an email address you hand out to a friend becomes spammed, changing
it requires you to inform your friend of the new address. SIP can
help solve this problem in part, by making use of presence [7].
Instead of handing out your email address to your friends, you would
hand out your presence URI. When a friend wants to send you an
email, they subscribe to your presence (indeed, they are likely
continuously subscribed from a buddy list application). The presence
data can include an email address where you can be reached. This
email address can be obfuscated and be of single use, different for
each buddy who requests your presence. They can also be constantly
changed, as these changes are pushed directly to your buddies. In a
sense, the buddy list represents an automatically updated address
book, and would therefore eliminate the problem.

3.8 Turing Tests

In email, Turing tests are those solutions whereby the sender of the
message is given some kind of puzzle or challenge, which only a human
can answer. If the puzzle is answered correctly, the sender is
placed on the user's white list. These puzzles frequently take the
form of recognizing a word or sequence of numbers in an image with a
lot of background noise. Automata cannot easily perform the image
recognition needed to extract the word or number sequence, but a
human user usually can. Since Turing tests rely on video or audio
puzzles, they sometimes cannot be solved by individuals with
handicaps.

Like many of the other email techniques, Turing tests are dependent
on sender identity, which cannot easily be authenticated in email.

Turing tests can be used to prevent IM spam, in much the same way
they can be used to prevent email spam. Indeed, the presence strong
authenticated identity techniques in SIP will make such a Turing test
approach more effective in SIP than in email.

Turing tests can be applied to call spam as well, although not
directly, because call spam does not usually involve the transfer of
images and other content that can be used to verify that a human is

on the other end. If most of the calls are voice, the technique
needs to be adapted to voice. This is not that difficult to do.
Here is how it could be done. User A calls user B and is not on user
B's white or black list. User A is transferred to an IVR system.
The IVR system tells the user that they are going to hear a series of
numbers (say 5 of them), and that they have to enter those numbers on
the keypad. The IVR system reads out the numbers while background
music is playing, making it difficult for an automated speech
recognition system to be applied to the media. The user then enters
the numbers on their keypad. If they are entered correctly, the user
is added to the whitelist.

This kind of voice-based Turing test is easily extended to a variety
of media, such as video and text, and user interfaces by making use
of the SIP application interaction framework [14]. This framework
allows client devices to interact with applications in the network,
where such interaction is done with stimulus signaling, including
keypads (supported with the Keypad Markup Language [15]), but also
including web browsers, voice recognition, and so on. The framework
allows the application to determine the media capabilities of the
device (or user, in cases where they are handicapped) and interact
with them appropriately.

In the case of voice, there are problems with the Turing test
described above. First, it is language specific. The application
could be made to run in different languages, if the caller indicates
their supported languages. This is possible in SIP, using the
Accept-Language header field, but this is not widely used at the
moment.

The other problem with this Turing test is the same one that email
tests have: instead of having an automata process the test, a spammer
can pay cheap workers to take the tests. Assuming cheap labor in a
poor country can be obtained for about $100 US dollars per year, and
assuming a Turing test of 30 second duration, this ends up being
about ten thousand messages per dollar, or about 10,000 microcents
per message. Though much more expensive than the 31 microcents per
message to send an IM spam, it is still relatively inexpensive.

As an alternative to paying cheap workers to take the tests, the
tests can be taken by human users that are tricked into completing
the tests in order to gain access to what they believe is a
legitimate resource. This was done by a spambot that posted the
tests on a pornography site, and required users to complete the tests
in order to gain access to content.

Due to these limitations, turing tests may never completely solve the
problem.

3.9 Computational Puzzles

This technique is similar to Turing tests. When user A tries to
communicate with user B, user B asks user A to perform a computation
and pass the result back. This computation has to be something a
human user cannot perform and something expensive enough to increase
user A's cost to communicate. This cost increase has to be high
enough to make it prohibitively expensive for spammers but
inconsequential for legitimate users.

One of the problems with the technique is that there is wide
variation in the computational power of the various clients that
might legitimately communicate. The CPU speed on a low end cell
phone is around 50 MHz, while a high end PC approaches 5 GHz. This
represents almost two orders of magnitude difference. Thus, if the
test is designed to be reasonable for a cell phone to perform, it is
two orders of magnitude cheaper to perform for a spammer on a high
end machine. Recent research has focused on defining computational
puzzles that challenge the CPU/memory bandwidth, as opposed to just
the CPU [19]. It seems that there is less variety in the CPU/memory
bandwidth across devices, roughly a single order of magnitude.

Recent work [21] suggests that, due to the ability of spammers to use
virus-infected machines (also known as zombies) to generate the spam,
the amount of computational power available to the spammers is
substantial, and it may be impossible to have them compute a puzzle
that is sufficiently hard that will not also block normal emails.
However, if combined with white listing, the computational puzzles
only become needed for validating new communication partners. The
frequency of communications with new partners is arguably higher for
email than for multimedia, and thus the computational puzzle
techniques may be more effective for SIP than for email in dealing
with the introduction problem.

These techniques are an active area of research right now, and any
results for email are likely to be usable for SIP. Of course, it is
likely that these techniques will come with a lot of patents and
other intellectual property constraints.

3.10 Payments at Risk

This approach has been proposed for email [20]. When user A sends to
user B, user A deposits a small amount of money (say, one dollar)
into user B's account. If user B decides that the message is not
spam, user B refunds this money back to user A. If the message is
spam, user B keeps the money. This technique requires two
transactions to complete: a transfer from A to B, and a transfer from
B back to A. The first transfer has to occur before the message can

be received in order to avoid reuse of "pending payments" across
several messages, which would eliminate the utility of the solution.
The second one then needs to occur when the message is found not to
be spam.

This technique appears just as applicable to call spam and IM spam as
it is to email spam. Like many of the other techniques, this
exchange would only happen the first time you talk to people. Its
proper operation therefore requires a good authenticated identity
infrastructure.

This technique has the potential to truly make it prohibitively
expensive to send spam of any sort. However, it relies on cheap
micro-payment techniques on the Internet. Traditional costs for
internet payments are around 25 cents per transaction, which would
probably be prohibitive. However, recent providers have been willing
to charge 15% of the transaction for small transactions, for
transactions as small as one cent. This cost would have to be
shouldered by users of the system. The cost that would need to be
shouldered per user is equal to the number of messages from unknown
senders (that is, senders not on the white list) that are received.
For a busy user, assume about 10 new senders per day. If the deposit
is 5 cents, the transaction provider would take .75 cents and deliver
4.25 cents. If the sender is allowed, the recipient returns 4.25
cents, the provider takes 64 cents, and returns 3.6 cents. This
costs the sender .65 cents on each transaction, if it was legitimate.
If there are ten new recipients per day, thats US $1.95 per month,
which is relatively inexpensive.

3.11 Legal Action

In this solution, countries pass laws that prohibit spam. These laws
could apply to IM or call spam just as easily as they could apply to
email spam.

There is a lot of debate about whether these laws would really be
effective in preventing spam. Whether they are or are not effective,
they would appear to be equally effective (or ineffective, as the
case may be) in preventing SIP spam.

As a recent example in the US, "do not call" lists seem to be
effective. However, due to the current cost of long distance phone
calls, the telemarketing is coming from companies within the US. As
such, calls from such telemarketers can be traced. If a telemarketer
violates the "do not call" list, the trace allows legal action to be
taken against them. A similar "do not irritate" list for VoIP or for
email would be less likely to work because the spam is likely to come
from international sources. This problem could be obviated if there

was a strong way to identify the sender's legal entity, and then
determine whether it was in a jurisdiction where it was practical to
take legal action against them. If the spammer is not in such a
jurisdiction, the SIP spam could be rejected.

There are also schemes that cause laws other than anti-spam laws to
be broken if spam is sent. This does not inherently reduce SPAM, but
it allows more legal options to be brought to bear against the
spammer. For example, Habeas <http://www.habeas.com> inserts
material in the header that, if a spammer inserted it without an
appropriate license, allegedly causes the spammer to be violating US
copyright and trademark laws, possibly reciprocal laws, and similar
laws in many countries.

3.12 Circles of Trust

In this model, a group of domains (e.g., a set of enterprises) all
get together. They agree to exchange SIP calls amongst each other,
and they also agree to introduce a fine should any one of them be
caught spamming. Each company would then enact measures to terminate
employees who spam from their accounts.

This technique relies on secure inter-domain authentication - that
is, domain B can know that messages are received from domain A. In
SIP, this is readily provided by usage of the mutually authenticated
TLS between providers. Email does not have this kind of secure
domain identification, although new techniques are being investigated
to add it using reverse DNS checks (see below).

This kind of technique works well for small domains or small sets of
providers, where these policies can be easily enforced. However, it
is unclear how well it scales up. Could a very large domain truly
prevent its users from spamming? Would a very large enterprise just
pay the fine? How would the pricing be structured to allow both
small and large domains alike to participate?

3.13 Centralized SIP Providers

In this technique, a small number of providers get established as
"inter-domain SIP providers". These providers act as a SIP-
equivalent to the interexchange carriers in the PSTN. Every
enterprise, consumer SIP provider or other SIP network (call these
the local SIP providers) connects to one of these inter-domain
providers. The local SIP providers only accept SIP messages from
their chosen inter-domain provider. The inter-domain provider
charges the local provider, per SIP message, for the delivery of SIP
messages to other local providers. The local provider can choose to
pass on this cost to its own customers if it so chooses.

The inter-domain SIP providers then form bi-lateral agreements with
each other, exchanging SIP messages according to strict contracts.
These contracts require that each of the inter-domain providers be
responsible for charging a minimum per-message fee to their own
customers. Extensive auditing procedures can be put into place to
verify this. Besides such contracts, there may or may not be a flow
of funds between the inter-domain providers.

The result of such a system is that a fixed cost can be associated
with sending a SIP message, and that this cost does not require
micro-payments to be exchanged between local providers, as it does in
Section 3.10. Since all of the relationships are pre-established and
negotiated, cheaper techniques for monetary transactions (such as
monthly post-paid transactions) can be used.

This technique can be made to work in SIP, whereas it cannot in
email, because inter-domain SIP connectivity has not yet been
established. In email, there already exists a no-cost form of inter-
domain connectivity that cannot be eliminated without destroying the
utility of email. If, however, SIP inter-domain communications get
established from the start using this structure, there is a path to
deployment.

This structure is more or less the same as the one in place for the
PSTN today, and since there is relatively little spam on the PSTN
(compared to email!), there is some proof that this kind of
arrangement can work. However, it puts back into SIP much of the
complexity and monopolistic structures that SIP promised to
eliminate. As such, it is a solution that the authors find
distasteful and contrary to the SIP design and architecture.

3.14 **Sender Checks**

In email, there has been a lot of interest in defining new DNS
resource records that will allow a domain that receives a message to
verify that the sender is a valid MTA for the sending domain [18]
[16].

Are these techniques useful for SIP? They can be used for SIP but
are not necessary. In email, there are no standards established for
securely identifying the identity of the sending domain of a message.
In SIP, however, TLS with mutual authentication can be used inter-
domain. A provider receiving a message can then reject any message
coming from a domain that does not match the asserted identity of the
sender of the message. Such a policy only works in the "trapezoid"
model of SIP, whereby there are only two domains in any call - the
sending domain, which is where the originator resides, and the
receiving domain. These techniques are discussed in Section 26.3.2.2

of RFC 3261 [2]. These techinques, however, are only applicable in
the trapezoid model where there is a sending and a receiving domain
only. In forwarding situations, the assumption no longer holds and
these techniques no longer work.

Thus, instead of creating DNS entries containing the IP address of
each legitimate relay for a domain, the provider can give each
legitimate relay a certificate that allows them to authenticate
themselves as coming from that domain. Such a technique would work
even in the face of IP address spoofing, which the marid techniques
are susceptible to.

4. Authenticated Identity in SIP

One of the key parts of many of the solutions described above is the
ability to securely identify the identity of a sender of a SIP
message. SIP provides a secure solution for this problem, and it is
important to discuss it here.

The solution starts by having each domain authenticate its own users.
SIP provides HTTP digest authentication as part of the core SIP
specification, and all clients and servers are required to support
it. Indeed, digest is widely deployed for SIP. However, digest
alone has many known vulnerabilities, most notably offline dictionary
attacks. These vulnerabilities are all resolved by having each
client maintain a persistent TLS connection to the server. The
client verifies the server identity using TLS, and then authenticates
itself to the server using a digest exchange over TLS. This
technique, which is also documented in RFC 3261, is very secure but
not widely deployed yet. In the long term, this approach will be
necessary for the security properties needed to prevent SIP spam.

Once a domain has authenticated the identity of a user, when it
relays a message from that user to another domain, the sending domain
can assert the identity of the sender, and include a signature to
validate that assertion. This is done using the SIP identity
mechanism [17].

A weaker form of identity assertion is possible using the P-Asserted-
Identity header field [6], but this technique requires mutual trust
among all domains. Unfortunately, this becomes expontionally harder
to provide as the number of interconnected domains grows. As that
happens, the value of the identity assertion becomes equal to the
trustworthiness of the least trustworthy domain. Since spam is a
consequence of untrusted domains and users that get connected to the
network, the P-Asserted-Identity technique becomes ineffective at
exactly the same levels of interconnectness that introduce spam.

A further weakness of P-Asserted-ID is that the actual domain which asserted the identity cannot be known. If that domain could be reliably known, then its assertions could be tempered based on user or domain-wide policiies. This weakness is not present in [17], which allows the recipient of a message to cryptographically determine the identity of the asserting domain.

SIP also defines the usage of TLS between domains, using mutual authentication, as part of the base specification. This technique provides a way for one domain to securely determine that it is talking to a server that is a valid representative of another domain.

5. Framework for Anti-Spam in SIP

Unfortunately, there is no magic bullet for preventing SIP spam, just as there is none for email spam. However, the combination of several techniques can provide a framework for dealing with spam in SIP.

Strong Authenticated Identity is Key: In almost all of the solutions discussed above, there is a dependency on the ability to authenticate the sender of a SIP message inter-domain. As such, we would argue that any provider that performs inter-domain SIP messaging must use the techniques described in Section 4, and in particular, depend on the strong identity techniques in [17].

Whitelists: With a strong identity mechanism in place, whitelists can facilitate communications from known callers. That reduces the scope of the problem to the introduction problem.

Consent Framework: The SIP consent framework [13] extends the presence framework for consent to all communications. Consent plays an important role in helping address the introduction problem.

Leverage What Email has to Offer: With the consent framework in place, spammers have only a small window through which they can introduce content to recipients. Fortunately, that problem is similar to traditional email spam, and can be addressed using the various email-based anti-spam techniques. Providers of SIP services should keep tabs on solutions in email as they evolve, and utilize the best of what those techniques have to offer. But perhaps most importantly, providers should not ignore the spam problem until it happens! That is the pitfall email fell into. As soon as a provider inter-connects with other providers, or allows SIP messages from the open Internet, that provider must consider how they will deal with spam.

6. Additional Work

Though the above framework serves as a good foundation on which to deal with spam in SIP, there are gaps, some of which can be addressed by additional work that has yet to be undertaken.

One of the difficulties with the strong identity techniques is that a receiver of a SIP request without an authenticated identity cannot know whether the request lacked such an identity because the originating domain didn't support it, or because a man-in-the-middle removed it. As a result, transition mechanisms should be put in place to allow these to be differentiated. Without it, the value of the identity mechanism is much reduced.

The consent framework depends on the ability for users to make a determination about whether to grant consent for unknown senders. In order for that framework to be useful, it needs to be coupled with techniques to ascertain trustworthiness. Reputation systems, for example, can help with that. At this time, reputation systems have seen implementation only within single domains, and using proprietary techniques. A standards-based inter-domain solution would be a valuable part of this framework.

7. Security Considerations

This memo is entirely devoted to issues relating to secure usage of SIP services on the Internet.

8. Acknowledgements

The authors would like to thank Rohan Mahy for providing information on Habeas, Baruch Sterman for providing costs on VoIP termination services, and Gonzalo Camarillo for his review. Useful comments and feedback were provided by Nils Ohlmeir, Tony Finch, Randy Gellens and Yakov Shafranovich.

9. Informative References

[1]   Campbell, B., "The Message Session Relay Protocol", draft-ietf-simple-message-sessions-10 (work in progress), February 2005.

[2]   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.

[3]   Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for

Instant Messaging", RFC 3428, December 2002.

[4]   Roach, A., "Session Initiation Protocol (SIP)-Specific Event
        Notification", RFC 3265, June 2002.

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

[6]   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.

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

[8]   Rosenberg, J., "A Watcher Information Event Template-Package
        for the Session Initiation Protocol (SIP)", RFC 3857,
        August 2004.

[9]   Rosenberg, J., "An Extensible Markup Language (XML) Based
        Format for Watcher Information", RFC 3858, August 2004.

[10]   Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource
        Identifiers (URI) Dynamic Delegation Discovery System (DDDS)
        Application (ENUM)", RFC 3761, April 2004.

[11]   Rosenberg, J., "The Extensible Markup Language (XML)
        Configuration Access Protocol (XCAP)",
        draft-ietf-simple-xcap-07 (work in progress), June 2005.

[12]   Rosenberg, J., "Presence Authorization Rules",
        draft-ietf-simple-presence-rules-02 (work in progress),
        February 2005.

[13]   Rosenberg, J., "A Framework for Consent-Based Communications in
        the Session Initiation Protocol (SIP)",
        draft-ietf-sipping-consent-framework-01 (work in progress),
        February 2005.

[14]   Rosenberg, J., "A Framework for Application Interaction in the
        Session Initiation Protocol (SIP)",
        draft-ietf-sipping-app-interaction-framework-04 (work in
        progress), February 2005.

[15]   Burger, E., "A Session Initiation Protocol (SIP) Event Package
        for Key Press Stimulus (KPML)", draft-ietf-sipping-kpml-07
        (work in progress), December 2004.

   [16]   Lyon, J., "Sender ID: Authenticating E-Mail",
          draft-ietf-marid-core-03 (work in progress), August 2004.

   [17]   Peterson, J. and C. Jennings, "Enhancements for Authenticated
          Identity Management in the Session Initiation Protocol (SIP)",
          draft-ietf-sip-identity-05 (work in progress), May 2005.

   [18]   Danisch, H., "The RMX DNS RR and method for lightweight SMTP
          sender authorization", draft-danisch-dns-rr-smtp-04 (work in
          progress), May 2004.

   [19]   Abadi, M., Burrows, M., Manasse, M., and T. Wobber, "Moderately
          Hard, Memory Bound Functions, NDSS 2003", February 2003.

   [20]   Abadi, M., Burrows, M., Birrell, A., Dabek, F., and T. Wobber,
          "Bankable Postage for Network Services, Proceedings of the 8th
          Asian Computing Science Conference, Mumbai, India",
          December 2003.

   [21]   Clayton, R. and B. Laurie, "Proof of Work Proves not to Work,
          Third Annual Workshop on Economics and Information Security",
          May 2004.

Authors' Addresses

   Jonathan Rosenberg
   Cisco
   600 Lanidex Plaza
   Parsippany, NJ 07054
   US

   Phone: +1 973 952-5000
   Email: jdrosen@cisco.com
   URI: http://www.jdrosen.net

   Cullen Jennings
   Cisco
   170 West Tasman Dr.
   San Jose, CA 95134
   US

   Phone: +1 408 527-9132
   Email: fluffy@cisco.com

      Jon Peterson
      Neustar
      1800 Sutter Street
      Suite 570
      Concord, CA 94520
      US

      Phone: +1 925 363-8720
      Email: jon.peterson@neustar.biz
      URI: http://www.neustar.biz

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment