Network Working Group
Request for Comments: 5039
Category: Informational

J. Rosenberg
C. Jennings
Cisco

January 2008

The Session Initiation Protocol (SIP) and Spam

Status of This Memo

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.

Table of Contents

1. Introduction

Spam, defined as the transmission of bulk unsolicited email, has been a plague on the Internet email system. 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. Consequently, it can be just as much of a target for spam as email. To deal with this, solutions need to be defined and recommendations put into place for dealing with spam as soon as possible.

This document serves to meet those goals by defining the problem space more concretely, analyzing the applicability of solutions used in the email space, identifying protocol mechanisms that have been defined for SIP that can help the problem, and making recommendations for implementors.

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. This is often called SPam over Ip Telephony, or SPIT.

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 that 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. This is often called SPam over Instant Messaging, or SPIM.

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 [6]), 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. This is occasionally called SPam over Presence Protocol, or SPPP.

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 User Agent Server (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 that 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 SIP User Agent 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 software 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 10 seconds 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 10 seconds of content.

With SIP, this cost is much reduced. Consider a spammer using a typical broadband Internet connection that provides 500 Kbps of upstream bandwidth. Initiating a call requires just a single INVITE message. Assuming, for simplicity's sake, that this is 1 KB, a 500 Kbps 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 5.3 Kbps), this requires approximately 16 Kbps after RTP, UDP, and IP overheads. With 500 Kbps upstream bandwidth, this means as many as 31 simultaneous calls can be in progress. With 10 seconds of content per call, that allows for 3.1 successful call attempts per second. If broadband access is around $50/month, the cost per successful voice spam is about 6.22 microcents each. This assumes that calls can be made 24 hours a day, 30 days a month, which may or may not be the case.

These figures indicate that SIP call spam is roughly four orders of magnitude 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 are few 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 URI 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 than 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 Voice over IP (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, 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.

It is useful to compare these figures with email. VoIP can deliver approximately 3.1 successful call attempts per second. Email spam can, of course, deliver more. Assuming 1 KB per email, and an upstream link of 500 Kbps, a spammer can generate 62.5 messages per second. This number goes down with larger messages of course. Interestingly, spam filters delete large numbers of these mails, so the cost per viewed message is likely to be much higher. In that sense, call spam is much more attractive, since its content is much more likely to be examined by a user if a call attempt is successful.

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 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, that is 62.5 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, most popular IM systems employ white lists, which only allow IM 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 and session mode. 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. The cost of this is within a small constant factor of IM spam so the same cost estimates can be used here. 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 [7]. 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. 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, while successful for email spam, is 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.

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@example.com) and entire domains (example.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, or zombies are used, black lists will have limited effectiveness for email.

Blacklists are also likely to be ineffective for SIP spam. Mechanisms for inter-domain authenticated identity for email and SIP are discussed in Section 4 and Section 5. Assuming these mechanisms are used and enabled in inter-domain communications, it becomes difficult 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 cannot 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, as described in Section 4.2 and Section 5, 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. Many popular IM systems also have strong identity mechanisms since they do not allow communications with IM systems in other administrative domains. 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. This is because SIP also provides a buddy list concept and has an advanced presence system as part of its specifications. The introduction problem remains. In email, techniques like Turing tests have been employed to address the introduction problem. Turing tests 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.

If a user's computer is compromised and used a zombie, that computer can usually be used to send spam to anyone that has put the user on their 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. Since most of today's popular IM systems only allow communications within a single administrative domain, sender identities can be authenticated. Email often uses similar consent-based systems for mailing lists. They use a form of authentication based on sending cookies to an email address to verify that a user can receive mail at that address.

This kind of consent-based communications has been standardized in SIP for presence, using the watcher information event package [7] and data format [8], which allow a user to find out that someone has subscribed. Then, the XML Configuration Access Protocol (XCAP) [10] is used, along with the XML format for presence authorization [11] to provide permission for the user to communicate.

A consent framework has also been developed that is applicable to other forms of SIP communications [12]. However, this framework focuses on authorizing the addition of users to "mailing lists", known as exploders in SIP terminology. Though spammers typically use such exploder functions, presumably one run by a spammer would not use this technique. Consequently, this consent framework is not directly applicable to the spam problem. It is, however, useful as a tool for managing a white list. Through the PUBLISH mechanism, it allows a user to upload a permission document [13] that indicates that they will only accept incoming calls from a particular sender.

Can a consent framework, like the ones used for presence, help solve call spam? 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 free form 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, it is possible to apply traditional content filtering systems to the header fields in the SIP messages, thus reducing these kinds of consent request attacks.

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, the value in sending spam is reduced, and perhaps it will cease for those spam cases where consent is not given to spammers.

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 ensure that 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. It may even be possible that an automated system could do some form of content analysis on this initial short message. 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 A attempts to contact user B. If a consent-based system is used, B is prompted to consent to communications from A, and along with the consent, 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 been successful in systems where centralization of resources (user identities, authentication, etc.) and monolithic control dominate. Examples of these include the large instant messaging providers that run IM systems that do not exchange messages with other administrative domains. 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 identity. Moreover, negative reputation is generated by tattling, which requires users to be annoyed enough to click the warning button -- a process that can be abused. In some reputation systems, "reputation mafias" consisting of large numbers of users routinely bully or extort victims by threatening collectively to give 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 artificial 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 acquire a new identity, since basing authorization decisions on positive reputation is essentially a form of white listing.

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 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 buddy lists would not affect your perception of their reputation unless their circle linked to your own social network.

If a users machine is compromised and turned into a zombie, this allows SPAM to be sent and may impact their reputation in a negative way. Once their reputation decreases, it becomes extremely difficult to reestablish a positive reputation.

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 minimize 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 Web sites 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 publicly accessible service containing addresses will suffice. As a result, ENUM [9] 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, each of which has constraints on its applicability. A limited-use address can be time-bound, so that it expires after a fixed period. Or, a different email address can be given to each correspondent. When spam arrives from that correspondent, the limited-use address they were given is terminated. In another variation, the same limited-use address is given to multiple users that share some property; for example, all work colleagues, all coworkers from different companies, all retailers, and so on. Should spam begin arriving on one of the addresses, it is invalidated, preventing communications from anyone else that received the limited use address.

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 [6].

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 to be 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.

Another approach is to give a different address to each and every correspondent, so that it is never necessary to tell a "good" user that an address needs to be changed. This is an extreme form of limited-use addresses, which can be called a single-use address. Mechanisms are available in SIP for the generation of [16] an infinite supply of single use addresses. However, the hard part remains a useful mechanism for distribution and management of those addresses.

3.8. Turing Tests

In email, Turing tests are mechanisms whereby the sender of the message is given some kind of puzzle or challenge, which only a human can answer (since Turing tests rely on video or audio puzzles, they sometimes cannot be solved by individuals with handicaps). These tests are also known as captchas (Completely Automated Public Turing test to tell Computers and Humans Apart). 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. The tests need to be designed such that automata cannot easily perform the image recognition needed to extract the word or number sequence, but a human user usually can. Designing such tests is not easy, since ongoing advances in image processing and artificial intelligence continually raise the bar. Consequently, the effectiveness of captchas are tied to whether spammers can come up with or obtain algorithms for automatically solving them.

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.

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 Interactive Voice Response (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 white list.

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, the Turing test would need to be made to run in the language of the caller. This is possible in SIP, using the Accept-Language header field, though this is not widely used at the moment, and meant for languages of SIP message components, not the media streams.

The primary problem with the voice 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 60 cents per hour, and assuming a Turing test of a 30-second duration, this is about 0.50 cents per test and thus 0.50 cents per message to send an IM spam. Lower labor rates would reduce this further; the number quoted here is based on real online bids in September of 2006 made for actual work of this type.

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 [26]. It seems that there is less variety in the CPU/memory bandwidth across devices, roughly a single order of magnitude.

Recent work [28] 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. If combined with white listing, computational puzzles would only be utilized for new communications partners. Of course, if the partner on the white list is a zombie, spam will come from that source. 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.

3.10. Payments at Risk

This approach has been proposed for email [27]. When user A sends email 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 make it arbitrarily 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, 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 0.75 cents and deliver 4.25 cents. If the sender is allowed, the recipient returns 4.25 cents, the provider takes 0.64 cents, and returns 3.6 cents. This costs the sender 0.65 cents on each transaction, if it was legitimate. If there are ten new recipients per day, that is US $1.95 per month, which is relatively inexpensive.

Assuming a micro-payment infrastructure exists, another problem with payment-at-risk is that it loses effectiveness when there are strong inequities in the value of currency between sender and recipient. For example, a poor person in a Third World country might keep the money in each mail message, regardless of whether it is spam. Similarly, a poor person might not be willing to include money in an email, even if legitimate, for fear that the recipient might keep it. If the amount of money is lowered to help handle these problems, it might become sufficiently small that spammers can just afford to spend it.

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.

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 it was inserted by a spammer without an appropriate license, would allegedly causes the spammer to violate 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 Transport Level Security (TLS)[22] between providers or SIP Identity [17].

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? At what point would the network be large enough that it would be worthwhile to send spam and 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

This technique is a variation on the circles of trust described in Section 3.12. 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 broadly 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, centralized architectures as these are deliberately eschewed because they put back into SIP much of the complexity and monopolistic structures that the protocol aims to eliminate.

4. Authenticated Identity in Email

Though not a form of anti-spam in and of itself, authenticated or verifiable identities are a key part of making other anti-spam mechanisms work. Many of the techniques described above are most effective when combined with a white or black list, which itself requires a strong form of identity.

In email, two types of authenticated identity have been developed - sender checks and signature-based solutions.

4.1. Sender Checks

In email, DNS resource records have been defined that will allow a domain that receives a message to verify that the sender is a valid Message Transfer Agent (MTA) for the sending domain [18] [19] [20] [21]. They don't prevent spam by themselves, but may help in preventing spoofed emails. As has been mentioned several times, a form of strong authenticated identity is key in making many other anti-spam techniques work.

Are these techniques useful for SIP? They can be used for SIP but are not necessary. In SIP, 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]. In forwarding situations, the assumption no longer holds and these techniques no longer work. However, the authenticated identity mechanism for SIP, discussed in Section 5, does work in more complex network configurations and provides fairly strong assertion of identity.

4.2. Signature-Based Techniques

Domain Keys Identified Mail (DKIM) Signatures [23] (and several non-standard techniques that preceded it) provide strong identity assertions by allowing the sending domain to sign an email, and then providing mechanisms by which the receiving MTA or Mail User Agent (MUA) can validate the signature.

Unfortunately, when used with blacklists, this kind of authenticated identity is only as useful as the fraction of the emails that utilize it. This is partly true for white lists as well; if any unauthenticated email is accepted for an address on a white list, a spammer can spoof that address. However, a white list can be effective with limited deployment of DKIM if all the people on the white list are those whose domains are utilizing the mechanism, and the users on that white list aren't zombies.

This kind of identity mechanism is also applicable to SIP, and is in fact, exactly what is defined by SIP's authenticated identity mechanism [17].

Other signature-based approaches for email include S/MIME[24] and OpenPGP[25].

5. Authenticated Identity in SIP

One of the key parts of many of the solutions described above is the ability to securely identify the sender of a SIP message. SIP provides a secure solution for this problem, called SIP Identity [17], 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 [5], but this technique requires mutual trust among all domains. Unfortunately, this becomes exponentially 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 the receiving domain not being able to trust the sending domains to disallow the hosts in the sending to send spam, the P-Asserted-Identity technique becomes ineffective at exactly the same levels of interconnectedness that introduce spam.

Consider the following example to help illustrate this fact. A malicious domain -- let us call them spam.example.com, would like to send SIP INVITE requests with false P-Asserted-Identity, indicating users outside of its own domain. spam.example.com finds a regional SIP provider in a small country who, due to its small size and disinterest in spam, accepts any P-Asserted-Identity from its customers without verification. This provider, in turn, connects to a larger, interconnect provider. They do ask each of their customers

to verify P-Asserted-Identity but have no easy way of enforcing it. This provider, in turn, connects to everyone else. As a consequence, the spam.example.com domain is able to inject calls with a spoofed caller ID. This request can be directed to any recipient reachable through the network (presumably everyone due to the large size of the root provider). There is no way for a recipient to know that this particular P-Asserted-Identity came from this bad spam.example.com domain. As the example shows, even though the central provider's policy is good, the overall effectiveness of P-Asserted-Identity is still only as good as the policies of the weakest link in the chain.

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.

6. 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. This section provides recommendations for network designers in order to help mitigate the risk of spam.

There are four core recommendations that can be made:

Strong Identity: Firstly, 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. Consent, reputation systems, computational puzzles, and payments at risk, amongst others, all work best when applied only to new requests, and successful completion of an introduction results in the placement of a user on a white list. However, usage of white lists depends on strong identity assertions. Consequently, any network that interconnects with others should make use of strong SIP identity as described in RFC 4474. P-Asserted-Identity is not strong enough.

White Lists: Secondly, with a strong identity system in place, networks are recommended to make use of white lists. These are ideally built off existing buddy lists, if present. If not, separate white lists can be managed for spam. Placement on these lists can be manual or based on the successful completion of one or more introduction mechanisms.

Solve the Introduction Problem: This in turn leads to the final recommendation to be made. Network designers should make use of one or more mechanisms meant to solve the introduction problem.

Indeed, it is possible to use more than one and combine the results through some kind of weight. A user that successfully completes the introduction mechanism can be automatically added to the white list. Of course, that can only be done usefully if their identity is verified by SIP Identity. The set of mechanisms for solving the introduction problem, as described in this document, are based on some (but not all) of the techniques known and used at the time of writing. 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.

Don't Wait Until It's Too Late: But perhaps most importantly, providers should not ignore the spam problem until it happens! 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.

7. 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.

8. Security Considerations

This document is entirely devoted to issues relating to spam in SIP and references a variety of security mechanisms in support of that goal.

9. 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 and Vijay Gurbani for their reviews. Useful comments and feedback were provided by Nils Ohlmeir, Tony Finch, Randy Gellens, Lisa Dusseault, Sam Hartman, Chris Newman, Tim Polk, Donald Eastlake, and Yakov Shafranovich. Jon Peterson wrote some of the text in this document and has contributed to the work as it has moved along.

10. Informative References

[1] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[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] 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.

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

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

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

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

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

[11] Rosenberg, J., "Presence Authorization Rules", RFC 5025, October 2007.

[12] Rosenberg, J., "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", Work in Progress, October 2007.

[13] Camarillo, G., "A Document Format for Requesting Consent", Work in Progress, October 2007.

[14] Rosenberg, J., "A Framework for Application Interaction in the Session Initiation Protocol (SIP)", Work in Progress, October 2005.

[15] Burger, E. and M. Dolly, "A Session Initiation Protocol (SIP) Event Package for Key Press Stimulus (KPML)", RFC 4730, November 2006.

[16] Rosenberg, J., "Applying Loose Routing to Session Initiation Protocol (SIP) User Agents (UA)", Work in Progress, June 2007.

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

[18] Allman, E. and H. Katz, "SMTP Service Extension for Indicating the Responsible Submitter of an E-Mail Message", RFC 4405, April 2006.

[19] Lyon, J. and M. Wong, "Sender ID: Authenticating E-Mail", RFC 4406, April 2006.

[20] Lyon, J., "Purported Responsible Address in E-Mail Messages", RFC 4407, April 2006.

[21] Wong, M. and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, April 2006.

[22] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[23] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and M. Thomas, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

[24] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[25] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

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

[27] 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.

[28] 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
Edison, NJ
US

EMail: jdrosen@cisco.com
URI: http://www.jdrosen.net

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

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

Full Copyright Statement

Intellectual Property