Consent Framework for Fighting Spam v0.3

Written by Yakov Shafranovich <research at solidmatrix dot com> , last updated on October 30th, 2003.

[PLEASE NOTE THAT THIS DOCUMENT IS NO LONGER BEING UPDATED OR SUPPORTED SINCE THE ASRG CHARTER HAS BEEN SWITCHED TO NON-CONSENT WORK. THIS DOCUMENT IS PROVIDED FOR HISTORICAL PURPOSES.]

Abstract

This document defines an abstract framework for consent-based communication systems. It defines what consent is, how consent is expressed, managed, and shared; different components of consent systems, services provided by these components and terminology used in consent systems. The goal of this document is to provide a framework for developing, evaluating, understanding and comparing anti-spam tools.

1. Introduction.

1.1. Need for a Model.

The current email system allows Internet users to exchange messages quickly and reliably. However in the recent years the problem of unwanted email messages, loosely referred to as “spam”, has increased in scale, growth, and effect; growing to account for a large percentage of the mail volume on the Internet. This unwanted traffic stands to affect local networks, the infrastructure, and the way that people use email. In [CHARTER] it was proposed to define a model for consent-based communications. In this section we outline why we think that such model is required.

1.1.1. Lack of a Single Definition.

Compounding the problem, the definition of spam messages is not clear and is not consistent across different individuals or organizations as mentioned by [CHARTER]. Most definitions tend to focus on two aspects of spam as illustrated by the definition provided by [SPAMHAUS]:

o Unsolicited - means that the recipient has not granted verifiable permission for the message to be sent

o Bulk - means that the message is sent as part of a larger collection of messages, all having substantively identical content

However as pointed out by [CONSIDER], format, content and traffic patterns of spam and non-spam may be identical, especially in the “bulk” aspect. It is very easy to recognize in practice the “bulk” aspect of spam or email which many distributed systems such as Razor have done, but it is very hard to recognize the “unsolicited” aspect of spam since that depends on the relationship between the email sender and the individual recipients. Therefore some spam while being “bulk” may be “solicited” and according to the above definition – not spam. However, the “unsolicited” aspect is very hard to prove especially considering that there is no standard way to grant or revoke a “verifiable permission”.

1.1.2. Interoperability.

The spam problem, its growth and effect on users has spawned a wide array of commercial and open-source tools to combat it. These range from filters such as SpamAssassin operating on a single machine to distributed spam detection networks spanning multiple servers such as DCC, Cloudmark and Razor. All of these tools continue to constantly evolve preempting and responding to spammers. According to [CONSIDER], this “evolution” process will continue and become a permanent part of the Internet together with spam. Therefore, according to [CONSIDER] in regards to spam, we seek to “reduce it to a tolerable level, rather than eliminate it.” Nevertheless, as the wide variety of anti-spam tools continues to increase, few efforts have been done to standardize protocols, formats, approaches and best current practices, so various anti-spam tools can communicate and be used in a component-based fashion.

1.1.3. Proposal Evaluation.

The spam problem has also generated considerable interest in the research community. However, currently there is no single abstract model for looking at the email system, spam, anti-spam tools and associated factors. This problem is especially relevant to our Anti-Spam Research Group (ASRG) which seeks to evaluate anti-spam proposals but lacks a model for evaluation. The notion of creating a model is not new to the Internet world, examples of existing models include instant messaging protocols in [RFC2778], policy management in [RFC3060] and content internetworking in [RFC3466].

1.2. Consent-Based Communications.

In [CHARTER] the problem of "unwanted email messages" is recast as a problem of "consent-based communication" and the model of anti-spam response is recast as a "consent model". According to [CHARTER] this means "that an individual or organization should be able to express consent or lack of consent for certain communication and have the architecture support those desires". The many anti-spam tools that are in use today can also be thought of as tools to enforce consent decisions made by users. When users define or configure anti-spam tools, they express their wishes as to what kind of email they want to receive and the anti-spam tools enforce those decisions. These wishes are what we refer to as "consent" to receive specific email. When users express their wishes as to what they want to receive and what they do not want to receive, they are expressing "consent" for certain types of email and are denying "consent" for certain types of email.

1.3. Purpose of This Model.

The current email system does not have the concept of "consent" built-in – email messages are delivered in a store and forward fashion without any provisions for rejecting email that lacks consent. Anti-spam tools are usually implemented as add-ons to existing email components to allow users to enforce those decisions. What we are seeking is to determine whether it is viable to have either a single architecture or a framework of components that can allow users to express and enforce their decisions as to what kind of email they want to receive – i.e. their consent decisions. We believe that such model can be helpful in reducing spam by allowing various tools to be combined providing a unified response, will allow users and organizations to use their own definitions of spam, and gives the ASRG an evaluation model that can be used for proposal evaluation.

1.4. Current Issues.

This section lists some of the current issues with the consent framework that need to be determined or worked out:

o Effectiveness on spam

o Scalability issues

o Deployment issues

o Whether the model is viable

o Whether a single format can encompass the wide range of consent decisions that users make

2. Overview.

The idea for consent model within the ASRG was proposed in [CHARTER]:

"*Therefore, we generalize the problem into "consent-based communication". This means that an individual or organization should be able to express consent or lack of consent for certain communication and have the architecture support those desires."*

In the consent model we would like to look at the spam problem and the response to that problem through the paradigm of "consent-based communications". As mentioned before, that means that users and organizations express their wishes for what kind of email they want to receive and what they do not want to the receive. The enforcement of those wishes by anti-spam and email tools make email into "consent-based communications" where email is sent, received and processed based on consent or revocation/denial of consent by users and organizations. While such abstract view encompasses the existing anti-spam tools and systems, we would also like to extend it further to tie in anti-spam tools together into one coherent whole with the email system in order to be able to provide a unified response to spam. In that case, the consent framework may be though of as a "system of systems" combining all anti-spam tools via standardized protocols into either one architecture or a modular framework.

In the consent framework, users and organizations express their wishes as consent and non-consent, which maybe implicit or explicit. Users and organizations also define their wishes in regards to email messages without prior consent or lack of consent in place – for example, how to deal with email strangers. Various anti-spam tools and systems store and enforce these decisions. Additional internal and external systems provide information that is used for enforcement – such as DNS-based blocklists, reputation systems, e-postage systems, etc. The sender may also provide information within the body or headers of the email messages that is used during the enforcement process. Examples of such information are challenge/response messages, e-postage tokens, trusted sender digital certificates or tokens, etc., all of which are collectively called "consent tokens". There are also components which track the origin of the message in order to deter spammers. They are collectively known as "source tracking

components" and may also serve a dual function by providing data for enforcement. An example of such component is SenderBase which tracks emails from individual IPs and organizations, which can be used for both purposes.

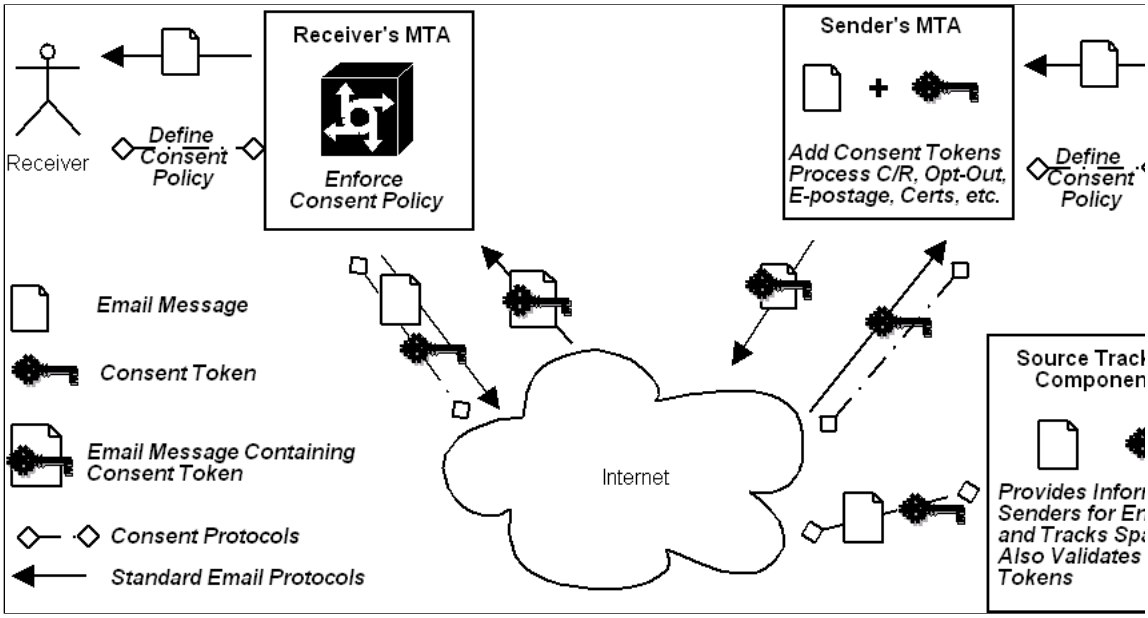

Figure 1. Overview of the Consent Model.

2.1. Consent and Revocation of Consent.

Individual email recipients, their ISPs and organizations make decisions about what kind of email they want to receive. The machine approximation of these human decisions are known within the model as "consent decisions", consisting of the following cases:

o Explicit Consent - users explicitly grant consent to a specific sender or a specific type of message. Example of these maybe whitelists of email addresses or IP addresses, trusted sender programs, Habeas, etc.

o Explicit Revocation of Consent – users explicitly revoke consent to a specific sender or a specific type of message. Examples of these are blacklists of email addresses or IP addresses, rejecting HTML email, filtering email for specific phrases, etc.

o Implicit Consent – users do not explicitly state what they want, and the software tools implicitly consents to the message based on its defaults, its settings and the consent decisions of the organization or ISP. Example of this is filtering done at the gateway level where the user does not have control. In respect to the user, consent has been granted or revoked implicitly since the user did not explicitly grant or revoke consent.

o Implicit Revocation of Consent – users do not explicitly state what they do not want. Software tools make the decision automatically for the user based on settings and consent decisions of the organization or ISP.

NOTE: We must keep in mind that consent and lack of consent expressed by user in actually vary greatly when put into a machine readable form due to the limitation of machines versus the human mind. Therefore, it is impossible to obtain a 100% correspondence between the original wishes of the user regarding consent and the machine expression of those wishes.

2.1.1. Lack of Prior Consent.

In many cases, especially in today's email system, total strangers may have the desire to communicate without any special arrangements. That situation is called "lack of prior consent". In most cases the decision on messages that lack prior consent is done implicitly by the system based on the settings provided by the user and/or organization. Depending on how the user or organization defines their consent policies, messaging lacking prior consent may be rejected outrighted, maybe filtered through a scoring system, maybe held up temporarily or may be let through. For example, a user may decide that all email without prior consent in plain ASCII should be allowed, while HTML email should not be allowed. This decision is left up to the user.

2.1.2. Expressing Consent via Consent Policies.

The various rules, and settings that users and organizations define are called "consent policies". Consent policies consists of a set of

rules and their associated data such as blacklists, whitelists, etc. They will be defined using a standard extensible format with a basic common set of consent rules defined as part of the format. Additional rules and settings will be defined via the extensibility extensions of the format. Standard default policies may also be defined as best current practices (BCP) documents.

2.1.3. Sharing of Consent Policies.

Consent policies are shared between the user, his or her organization or ISP. The organization and ISP may also have their own consent policies which are combined with the users' policies. All policies may include scope rules which define how the policy may and may not be shared. Certain parts of the consent policies may also be shared with the sender under specific circumstances – for example in challenge / response, a certain section of the consent policy may be sent as part of the challenge allowing the sender to pick a correct format for the response.

2.1.4. Consent Tokens.

Various tokens that are exchanged between different parts of the model are called “consent tokens”. These include special headers, digital certificates, e-postage stamps, and other similar devices, used by the sender when sending the original email message. They also include tokens such as challenge / response that are exchanged between the sender and receiver after the email message has been sent. They also encompass “opt-in” and “opt-out” decisions as tokens that are communicated by the receiver to the sender via a machine readable format.

Standard formats will be defined for consent tokens to cover most of current anti-spam systems. Additional token formats will be defined in the future as part of the evolution process of the consent framework. A registry of standard consent token types will probably be operated by IANA.

2.2. Model Components.

Based on [CHARTER] the consent framework consists of three types of components: consent expression components, policy enforcement components and source tracking components. Consent expression components let receivers create consent policies. Policy enforcement components enforce consent policies of users and organizations on incoming email. Source tracking components identify and track senders that violate the consent policies. They may also provide data about the the senders which may be used by the policy enforcement components.

2.3. Protocols.

1. MAIL TRANSFER PROTOCOL - used to transfer email messages between the SENDER and the RECEIVER. This transfer process may involve one or more software AGENTS.

2. CONSENT POLICY EXCHANGE PROTOCOL (CPEP) – protocol for sharing CONSENT POLICIES.

3. CONSENT TOKEN EXCHANGE PROTOCOL (CTEP) – protocol for sharing CONSENT TOKENS.

4. SOURCE TRACKING EXCHANGE PROTOCOL (STEP) – protocol for sharing information about email sources and SENDERs.

NOTE: Please note that these protocols may be all defined as one single protocol with different sections for different needs.

2.4. Formats.

1. CONSENT POLICY DEFINITION LANGUAGE (CPDL) – for expressing SENDER, SCOPE and CONSENT POLICIES.

2. CONSENT TOKEN DEFINITION LANGUAGE (CTDL) – for expressing CONSENT TOKENS.

3. SOURCE TRACKING DEFINITION LANGUAGE (STDL) – for expressing information about email sources and SENDERs.

3. Model Terminology.

AGENT – a software tool that is representing a human user

CONSENT – expression of wanting to receive email from a specific SENDER, it may be EXPLICIT or IMPLICIT.

CONSENT EXPRESSION COMPONENT – components that lets RECEIVERS define CONSENT POLICIES

CONSENT POLICY – a combination of rules defining what types of email the RECEIVER has expressed CONSENT and DENIED CONSENT. It also includes a SENDER POLICY and a SCOPE POLICY. It may be defined for a specific RECEIVER such as an individual user or a group of RECEVIERS such as an organization or an ISP. CONSENT POLICIES may be shared and may incorporate or include parts of other CONSENT POLICIES.

CONSENT POLICY DEFINITION LANGUAGE (CPDL) – for expressing SENDER, SCOPE and CONSENT POLICIES.

CONSENT POLICY EXCHANGE PROTOCOL (CPEP) – protocol for sharing CONSENT POLICIES.

CONSENT TOKEN – a piece of data that is used by the POLICY ENFORCEMENT COMPONENT to grant or deny CONSENT based on the CONSENT POLICY.

CONSENT TOKEN DEFINITION LANGUAGE (CTDL) – for expressing CONSENT TOKENS.

CONSENT TOKEN EXCHANGE PROTOCOL (CTEP) – protocol for sharing CONSENT TOKENS

DENIED CONSENT – expression of not wanting to receive email from a specific SENDER, it may be EXPLICIT or IMPLICIT.

EXPLICIT – result of an explicit rule set by the RECEIVER

IMPLICIT – result of an evaluation by some mechanism chosen by the RECEIVER such as an email filter

MAIL TRANSFER PROTOCOL – a protocol used to transfer email between the SENDER and the RECEIVER

POLICY ENFORCEMENT COMPONENT – a component that enforces CONSENT POLICIES for the RECEIVER.

RECEIVER – the owner of the email address to which the email message is addressed.

SCOPE POLICY – a combination of rules defining how a CONSENT POLICY may be shared, and with whom it may be shared.

SENDER - defined by a combination of characteristics of the sender such as IP, email route, email address, or some authentication mechanisms. This definition is specified by the SENDER POLICY. The RECEIVER may define any of these for a particular sender, and may also define which authentication mechanisms (if any) are used. Authentication may be performed by presented a CONSENT TOKEN or by input from an SOURCE TRACKING COMPONENT.

SENDER POLICY – a combination of rules defining a SENDER

SOURCE TRACKING COMPONENT – components used to identify and track SENDERS either to help make POLICY ENFORCEMENT COMPONENT make decisions or to deter SENDERS from violating the CONSENT POLICY

SOURCE TRACKING DEFINITION LANGUAGE (STDL) – for expressing information about email sources and SENDERs

SOURCE TRACKING EXCHANGE PROTOCOL (STP) – protocol for sharing information about email sources and SENDERs

4. Security Considerations.

[Ed. To be added]

5. Privacy Considerations.

[Ed. To be added]

6. Acknowledgments.

This document is the work of the Anti Spam Research Group (ASRG) of the Internet Research Task Force (IRTF). The author would like to acknowledge the valuable input of all members of the group especially Paul Judge, Andrew Akehurst, Walter Dnes, Gordon Peterson, John Kyme, John Fenley, Jonathan Morton and Peter McNeil.

7. References.

[CHARTER] Charter of the ASRG, current version available at (http://www.irtf.org/charter?gtype=rg&group=asrg)

[SPAMHAUS] The SpamHaus organization, website (http://www.spamhaus.org)

[CONSIDER] Crocker. D., Schryver, V. and Levine, J.; “Technical Considerations for Spam Control Mechanisms”, Internet draft, May 2003

[RFC2778] Day, M., Rosenberg, J., Sugano, H.; “A Model for Presence and Instant Messaging”, RFC 2778, February 2000

[RFC3060] Moore, B., Ellesson, E., Strassner, J., Westerinen, A.; “ Policy Core Information Model -- Version 1 Specification”, RFC 3060, February 2001.

[RFC3466] Day, M., Cain, B., Tomlison, G., Rzewski, P.; “A Model for Content Internetworking (CDI)”, RFC 3466, February 2003

Appendix A – Document History

0.1 / July 4th, 2003 / Initial publication

0.2 / July 5th, 2003 / Took out non-essential stuff, rewrote certain portions

0.3 / October 30th, 2003 / Complete rewrite of most sections except glossary, added a diagram

---