

Mail Message Metering

Originally published in *_;login: The Magazine of USENIX and SAGE_ (December 2000)*. Published by permission of The USENIX Association

Problem Definition

The scourge of spam has existed since the commercialization of the Internet began in the early nineteen nineties. Unfortunately, spam is here to stay irrespective of any legal, procedural and technical measures are used to contain it. ZipLink, as a wholesale provider of Internet access, is in a unique position to do its part in stopping spam from emanating from its network saving other network operators and end users time, money and inconvenience.

The March 6, 2000 edition of *Interactive Week* contained an article outlining the bankruptcy of AGIS, a wholesale provider of Internet access who harbored spammers. In this article the author Max Smetannikov states, "(Lawlor, president of AGIS) opened AGIS to unsolicited commercial e-mailers and only relented after a walkout of key technical staff and a crippling hack attack in 1997." The issue of spam is critical for organizations like ZipLink. If we don't try to eliminate spam, our transit providers and peers will shut us down, causing our business to cease operating as almost happened to AGIS.

The problem of unsolicited commercial email or UCE from a wholesale provider's perspective is two part:

- how to reroute all mail traffic to a single point (mail relay) without causing undue stress on our wholesale customers and end subscribers, and
- how to identify a bulk mailer in progress

The first problem is solved by redirecting SMTP (outbound mail) sessions to centralized mail relays at appropriate points throughout our network. The second issue depends on how one defines a spammer. ZipLink believes that a user who exceeds a certain message count per unit of time can be accurately defined as a spammer in progress.

Our goals for a technical solution for the problem of spam include:

- blocking at least 50% of outbound spam;
- creating little or no impact on customer ISP and end user;
- having the ability to exclude certain domains and users;
- providing a solution that is configurable and scalable;
- minimizing the impact on existing infrastructure; and
- utilizing free software where possible.

Of course, blocking spam is the most important goal. While in production numbers don't yet exist, we believe 80% of all outbound spam can effectively be blocked with our unique method. Limiting the impact on the customer ISP and end subscriber is important. As a result, our solution includes the use of layer 4 switches to automatically redirect SMTP connections to centralized mail relays without intervention by ZipLink, customer ISP or the end user.

Since many of our ISP customers do a very good job at blocking UCE on their own and have explicitly asked to be excluded from our anti-spam solution, any anti spam filtering method put in place must include an opt-out on a customer by customer (ISP) basis. Also, some end subscribers have legitimate reasons for sending large amounts of email that may appear to an observer as UCE. These bulk mailers must be able to send their legitimate mail on an ongoing basis without continuous intervention.

Of course, any solution cannot render our existing infrastructure unusable, so it must have little impact on our RADIUS (Remote Authentication Dial In User Services, defined by RFC2138 and RFC2139) servers. Lab testing indicates an impact of approximately 5% on our RADIUS servers, which is well within our

goal. Finally, it is important to leverage existing solutions in order to reduce development time and cost, if at all possible.

Existing Solutions for Spam

The existing solutions for curtailing spam fall into four general categories:

- Services, such as Brightmail and MAPS' RBL
- Client based solutions, such as "POP before SMTP" and Hash Cash
- Server based solutions, such as Blackmail and SpamShield
- RAS filters

Service based solutions are aimed squarely at the end user or ISP providing end user services, which ZipLink does not provide as a wholesaler of Internet access. Client based solutions are too expensive in terms of end user support costs and don't meet our requirement for minimal impact on end users. Existing server based solutions do not have the required flexibility and are aimed again at the receipt of spam, not for sources of spam such as wholesale access providers.

Many wholesale providers today, including ZipLink, utilize RAS filters to block UCE. While such filters work, they have at least three major limitations:

- filters are not manageable with 500 ISP's (since at least 1 unique filter is required per ISP)
- filters significantly decrease dial up performance of the RAS equipment and therefore can impact the customer's online experience significantly, and
- RAS equipment has finite memory available for filters which limits the number of filters which can be applied across the network

These shortcomings severely limit the usefulness of filters on RAS equipment as a method to block spam. However, they do serve a purpose and will likely be used for certain customers (free/subsidized ISP's, for example) in the foreseeable future.

Mail Message Metering

Under ZipLink's patent pending method of blocking spam, UCE is defined as large volumes of messages sent over a short period of time. Spammers don't normally send a message per hour; they usually try to send as many messages as they possibly can over the shortest period of time.

Figure one shows a block diagram of our solution in its entirety. The basic premise of the system is that a particular IP address may originate only X number of messages during Y amount of time. A message is defined as a to: or cc: recipient in the header of the message. For example, if the user had a to: line with three addresses and a cc: line with three addresses, a total of six messages would be charged against the user's quota. In order to allow exceptions to the defined maximums, we track the user assigned to the IP address via RADIUS accounting records. RADIUS records are placed into an Oracle database as the same record is concurrently entered into the usual RADIUS accounting file for redundancy purposes.

To illustrate exactly how the process works, it is useful to track the path of a mail message through the system. As an end subscriber initiates an outbound mail connection, a layer four switch redirects the session to a shadow SMTP server. That mail server performs a quota lookup in the Oracle database and determines whether or not this user has exceeded their user, domain, global or compiled in (default) limit. If the user's quota has not been exceeded, the message counts in the Oracle database for that user and domain are updated and the message is allowed to pass. If the user has exceeded their limits, the customer receives a "450 Mail quota exceeded for %U" message.

The layer four switch contains configuration logic that can except certain domains from being forwarded to the quota checking mail relays. As a result, ISPs/domains who do a good job and don't cause spamming

problems can easily be excepted from the system. Of course, the ISP can easily be added back should they stop handling UCE appropriately.

Getting RADIUS Records into the Oracle database

It is worth noting here that one could easily forego the saving of RADIUS data (or other IP address to user mapping) into an Oracle database if one didn't have such data. The result of doing this would be an inability to allow certain end users or domains to exceed default values. In addition, the RADIUS data could easily be replaced with DHCP or static IP address data. The reason RADIUS data is used is due to the fact that our RAS equipment utilizes RADIUS for authentication, authorization and accounting as do most (if not all) ISPs.

Figure 2 outlines how data enters the Oracle database. Each RADIUS server runs a program called "radius2db" which is a series of Oracle function calls that forwards RADIUS data into Oracle. This program is approximately 800 lines of C code and was written entirely in house by Dale Nielsen. A mysql version of the code exists, and should ZipLink decide to open source this solution, this version will likely be released.

In our preliminary testing, we have shown that the impact on our RADIUS servers meets our requirements for limited impact. In fact, the process related to this activity takes approximately 5% on both a dual processor 300 Mhz Sun Ultra 2 with 768 MB of memory, and a 300 Mhz Sun Ultra 5 with 512 MB of memory. The impact on our Oracle database will be tested once we go into limited production testing. Since Oracle is relatively easy to scale, impact on it is less important. It is expected that the existing Oracle infrastructure (a Sun E450 with a single processor and 1 GB memory) will easily be able to handle 100,000 ports of traffic, which is approximately double the size of our existing network.

Quota Checking Mail Relays

When we evaluated solutions for our quota checking mail relays, we closely examined two firewall proxy SMTP solutions: the Trusted Information Systems (TIS) Firewall Toolkit SMAPD and the Obtuse Systems Corporation Juniper Firewall Toolkit SMTPD. While very similar in functionality and features, we picked SMTPD from the Juniper Firewall Toolkit for the following reasons:

- The license is "BSD style" and therefore less encumbered than the TIS toolkit, and
- SMTPD runs as a daemon which we feel is more reliable than running out of inetd as SMAPD does.

SMTPD performs the following functions on each mail message:

- verifies source IP address is within the ZipLink range,
- checks mail quotas for that user, and
- forwards message to SMTPFWD.

SMTPFWD, part of the Juniper Firewall Toolkit, simply utilizes sendmail to forward the message to the ISP customer's mail relay. Although each ISP customer's mail relay needs to be entered into the sendmail configuration, no changes were made to the sendmail source code. The failure mode of the quota checking mail relays is to allow the message to pass if it cannot reach the Oracle database.

Tunable Parameters

In order to enable per domain and per user limits, quotas must be kept in the Oracle database. The quotas are checked in the following order:

1. number of messages per time interval for user@realm
2. number of messages per time interval for @realm
3. number of messages per time interval, global
4. 10 messages/10 minutes, max 100 messages/24 hours is the compiled in default

If the first limit (number of messages per time interval for [user@realm](#)) doesn't exist, then the check drops to the next limit. For each level, there are two sets of checks done. For example, there might be a limit of 100 messages per 60 minutes and 1000 messages per 24 hours for [user@isp.net](#). If either one of the limits is exceeded, then the message is not forwarded. If either quota has not been met, then the message counts for that user and domain, the quota is updated to include the message and the message is forwarded on to the customer ISPs mail relay for final delivery to the end recipient.

For the majority of subscribers, the ISP can simply set a domain limit and that is all. For legitimate bulk mailers (such as opt in mailing lists) the ISP can set that particular customer's limits higher to an appropriate threshold. The customer, the customer's ZipLink account manager, or the ZipLink NOC can make changes to limits via a simple web browser interface.

Limitations

As with any complex system, nothing is perfect. Having a solution like this for the problem of UBE, it might be tempting to eliminate anti-UCE language in the Acceptable Use Policy (AUP). However, this system does still require an AUP. In fact, it would be a good idea if all ISP's placed message count limits directly in their AUP. This might go a long way to reduce UCE.

Another problem is that the limits set within the system can be circumvented. This might be the case if the ISP customer was sympathetic to their customers who generate UCE intentionally for profit. The easy fix to this is to not allow such ISPs to change their customers' limits.

One possible problem with this system is that end customers cannot relay mail through arbitrary mail servers. An example of this use would be an employee of a company working from a home office who would like to relay mail through his companies mail server. We're not sure if this poses a problem or not, but if it does, steps can be taken to enable certain customers this ability. This may require modifications to the sendmail code, but further analysis is required in order to complete a design.

One of the toughest problems is the fact that this design requires the use of layer 4 switches to redirect SMTP sessions to the quota checking mail relays. This is not a trivial problem for providers the size of ZipLink whose networks have not been engineered for this functionality. This will require the re-engineering of every ZipLink point of presence to ensure all traffic goes through a central point.

The final issue that we see with this system is that pro-First Amendment organizations such as the Center for Democracy and Technology and the Electronic Frontier Foundation may very well argue that we are impeding on our customer's First Amendment rights. As was stated previously, the problem of UCE is a matter of business continuity for most ISP's and must be addressed.

References

Trusted Information Firewall Toolkit: <http://www.tis.com/research/software>

Blackmail: <http://bitgate.com/spam>

Obtuse Systems Corporation Juniper/smtpd Firewall Toolkit: <http://www.obtuse.com/smtpd.html>

Brightlight Technologies Brightmail: <http://www.brightlight.com/isp/spam>

Mail Abuse Prevention System (home of RBL/TSI/DUL/RSS): <http://mail-abuse.org>

Hash cash: <http://www.cypherspace.org/~adam/hashcash/>

Spamcop: <http://spamcop.net>

Spamshield: <http://spamshield.conti.nu/>

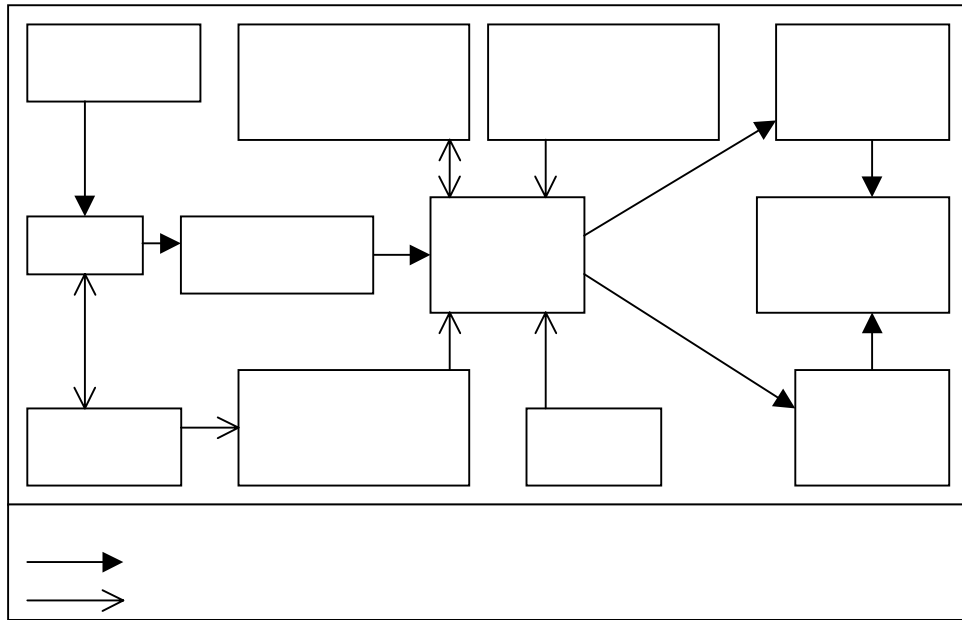


Figure 1

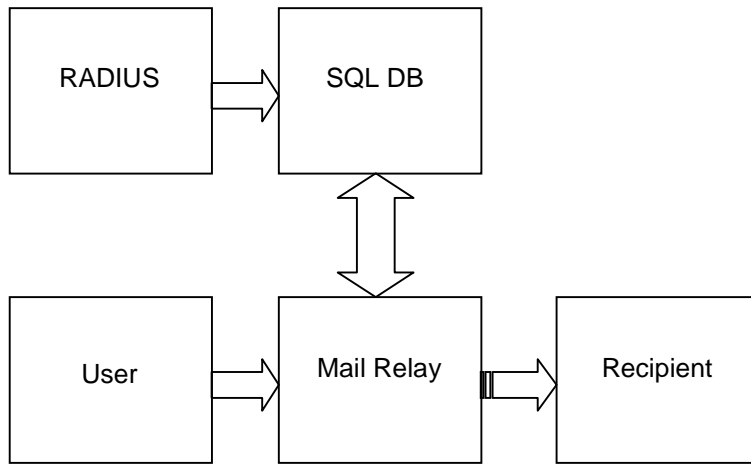


Figure 2