Experiences with Greylisting

John R. Levine
Taughannock Networks, PO Box 727, Trumansburg NY 14886

Abstract

Greylisting temporarily rejects mail from unknown sources on the theory that real mailers will retry while spamware won't. I outline a taxonomy of greylisters and report some statistics both on anti-spam effectiveness and its effect on non-spam mail.

1 Why Greylisting?

For many years, large amounts of spam has been sent through purpose-built spamware, rather than normal MTAs. If recipient hosts can identify distinctive characteristics of spamware that differ from legitimate MTAs, the recipient hosts can reject mail from spamware during the SMTP session, avoiding the need to receive the spam.

Spamware consistently does little or no error recovery. If it can't deliver a message, it just goes on since in spamming, volume counts for far more than reliability. Greylisting tries to detect spamware by rejecting mail from unfamiliar sources with a soft fail 4xx) error code, on the theory that real MTAs will retry, and spamware won't. Another, less well developed application of greylisting is to delay mail from newly seen IP addresses on the theory that if it's a spam source, even if it retries, it'll appear on a blacklist before the mail is accepted.

2 Flavors of greylisting

Greylisting has been independently invented and implemented several times[3,4,5]. Although all versions do approximately the same things, details differ in interesting ways that can affect its accuracy and performance.

The first detail is what mail is greylisted. One could greylist everything, but there's little point to doing so. Typical greylisters maintain a whitelist of IP addresses known to retry. They greylist mail from new IP addresses, and add an IP to the list if it retries.r2 The whitelist invariably also has manually maintained entries, for senders that don't deal well with being greylisted. Harris' initial greylist prototype[3] greylists on every unique triple of (IP, sender, recipient) and keeps a whitelist of triples, not IP addresses. Networks with multiple servers generally use one shared greylist database with a shared whitelist and usually a shared list of retryable deliveries.

The next detail is just what does retry mean, and how does the system recognize a new message as a retry of a rejected one. Greylisters can remember a variety of data about messages: sending IP address, recipient IP address, message envelope, bounce and recipient addresses, and the message body, either with the `Message-ID` or a checksum of the message body.

The third detail is where in the SMTP process the greylisting happens. Possibilities include: after the RCPT TO, immediately after DATA, or after the message has been received. The farther into the SMTP session the greylisting happens, the more exactly the server can match up retries with the original message, which may or may not be a good idea.

3 How greylisting fails

Greylisting can fail in two ways. The more serious way is that it loses legitimate mail. The less serious is that it needlessly delays legitimate mail. Although some spam will retry and get through, we don t characterize that as a failure.

3.1 Losing messages

Since the 4xx failure codes that greylisting issues are the same as a SMTP server would issue if it were overloaded, any correct client should retry. Some hosts just don't retry, either by policy or due to bugs. Yahoo Groups is the largest source that hasn't retried in the past, probably due to performance limits, although recently they have started to do so. A few mail client packages don't handle 4xx messages correctly, either at all or at certain points in the SMTP transaction. In particular, if the server returns 4xx at the end of data,

a lot of clients will act as though it were a 5xx and bounce the message back to the sender. In my experience, these systems are rare enough that I've manually whitelisted the few I've seen.

Matching retries to the corresponding original message is more difficult. Sending systems with a pool of servers frequently retry from a different server, which causes trouble for greylisters that use the sending IP address as a key. Mailing list software frequently uses a unique bounce address per recipient, per message, and occasionally per delivery. This causes trouble for greylisters that key on the bounce address. A few systems regenerate the message anew for each delivery attempt. This causes trouble for greylisters that key on the message checksum.

My experience suggests that matching messages by (IP, sender, recipient) with IP whitelisting works well, although detailed analysis of the data shows legitimate sending IPs that never get whitelisted due to retry patterns. Heuristics to identify retries from different IPs could make the whitelist more complete. Entries expire from the whitelist after sending no mail for a long time, originally 7 days, since raised to 30 days with a marginal improvement in whitelisting and no effect on accuracy.

3.2 Delays and unfortunate interactions

If a mail server can tell during an initial message delivery that the client would retry after a 4xx, there's no point in greylisting. Also, if the server can identify mail addressed to software processes rather than people, such as mailing list bounces, it might as well accept it since the processes can ignore spam without human help. My majordomo2 list manager uses easily recognizable bounce addresses, so I added special case code never to greylist them. Greylisting can interact in unfortunate ways with techniques such as VERP[1] and BATV[2] that encode information into bounce addresses. VERP (Variable Envelope Return Paths) encodes recipient information into the bounce address, which makes a mailing list appear to be many different senders, causing trouble for systems that whitelist on (IP, sender, recipient) rather than IP.

BATV puts a signature and time stamp in each bounce address, to help detect bounces due to mail forged by third parties which won't have the signatures. The time stamp limits attacks via addresses scraped from archives. If BATV computes the signature at delivery time and a message soft fails and is retried, the new signature may have a different time stamp, hence a different bounce address. To avoid that problem the time stamp could reflect the time the message was queued rather than delivery time. In practice it hasn't been a problem because the granularity of the time stamp is a day, much longer than the typical delivery retry interval.

4 Statistics

I analyzed my greylist package's logs for a seven-week period in 2005. It handles two servers that contain a mixture of several hundred individual mailboxes, a few dozen mailing lists, and the abuse.net message forwarding system. Of a total of 715,000 delivery attempts, 11% were to manually whitelisted addresses, 62% were to addresses that had retried before and been whitelisted automatically, and 20% were greylisted. Of the greylisted addresses, only 16% retried successfully. Most successful retries happened within a few minutes, with clusters at 400 and 900 seconds, probably representing the retry time of popular MTAs.

Under 2% of mail to whitelisted addresses and 4% of successfuly retried deliveries had more than one recipient, compared to 35% of those that never retried and 8% that retried too soon. Most greylisted attempts that had three or more recipient addresses never retried, or retried so much later that it wasn't recognized as the same message evidently an artifact of a spamware package.

5 Summary

Greylisting is a successful approach to rejecting spam sent by slopplily implemented spamware. However, greylisting has can reject significant amounts of legitimate mail, due both to sending MTAs that don't retry, and difficulties in recognizing valid but unusual retry techniques. Careful system design can minimize the amount of lost legitimate mail with little loss of effectiveness against spam.

References

[1] Dan Bernstein (1997), *Variable Envelope Return Paths*, http://cr.yp.to/proto/verp.txt.

[2] John Levine et al.(2004), *Bounce Address Tag Validation*, http://www.mipassoc.org/batv/index.html.

[3] Evan Harris, *The Next Step in the Spam Control War: Greylisting*, http://projects.puremagic.com/greylisting/whitepaper.html.

[4] Vern Schryver (2005), *Distributed Checksum Clearinghouse*, http://www.rhyolite.com/anti-spam/dcc/.

[5] Emmanuel Dreyfus, *milter-greylist for Sendmail*, http://hcpnet.free.fr/milter-greylist/.