Eliminating spam and enabling email privacy through the use of Programmable Email Addresses

August 3, 2004

Peter Kay, President

Titan Key Software, LLC

Honolulu, Hawaii

peter.ceas@titankey.com

Abstract

Today's email paradigm unconditionally delivers properly addressed email. Subsequently, the great majority of today's spam elimination techniques is reactive and so far has not resulted in abating the spam problem. A technology allowing the end-user to create multiple, *programmable* email addresses (PEAs) shatters the paradigm by *conditionally* accepting email. Only email from senders who have been approved against a collection of policies and databases are accepted. Disapproved email is rejected before the DATA command is accepted. As a result of using PEAs, the end user has complete control over their email privacy. By creating a new paradigm of *conditional email acceptance,* the end-user enjoys a spam-free inbox, no false positives, and no quarantine folders to review. And because unwanted email is stopped *before it can be sent,* all bandwidth infrastructure costs related to spam drop to near-zero.

Overview

We have spam because we have no privacy

The root cause of spam stems from the fact that from the beginning of SMTP, *all properly addressed email is unconditionally accepted.* The moment a user gives out their email address is the moment they lose any ability to control email traffic to their inbox. The lack of privacy ultimately leaves the end-user in a powerless, reactive condition in dealing with unwanted email.

Spam cannot be stopped if it has to be accepted

The great majority of anti-spam technologies are based on some type of filtering. Simply put, if a technology requires the receipt and processing of spam in order to detect it, *by definition, it cannot stop it*. This paper describes a technology that for the first time allows an end-user to dynamically bind a policy to a given email address and any senders violating that policy are stopped *before they can send the email contents.*

Introducing the Programmable Email Address (PEA)

PEAs allow the end-user to create a collection of email addresses for specific purposes while allowing the needs and policies of those email addresses to change over time. Unlike filtering, email that violates end-user policy is stopped before the contents are sent. Unlike DEAs (Disposable Email Addresses) which typically either only accept or reject all email, a PEA can continue accepting email from some while rejecting email from others. Unlike recent DEA improvements, such as the recent AT&T "SPA" (Single Purpose Address), commercialized by zoEmail.com, a PEA policy can be dynamic and change over time.

Because PEA technology makes use of databases and policies, they are "programmable", meaning that they can be given a collection of rules and then act on those rules over time and email volume. Policies can be defined upon creation, dynamically altered based on incoming email characteristics, or changed over time based on the needs of the users. PEAs can enforce new email protocols for a very specific subset of the user's total email inflow, allowing new standards to be introduced without requiring an all-or-nothing approach to adopting new standards.

Once a user has acclimated to the concept of using a collection of PEAs for various purposes, their inbox is free of spam, there is no quarantine folder to review, and they have unprecedented control over who can send them email. PEA's completely eliminate spam, solve the email privacy problem, remain compatible with today's SMTP standards, and allow incremental implementation of new email protocols.

How PEA's Work

PEAs look and work like normal email addresses. What differentiates them is how they are processed by their host server. Unlike all other mail servers which unconditionally accept properly addressed email to a working email address, a PEA conditionally accepts email based on the specific policy it is bound to. This policy is examined and enforced during the early stage of an SMTP transaction. Transactions that attempt to violate the policy are stopped with a fatal error before accepting the email contents. Because PEAs store their policy and usage into a relational database, policies can be flexible and change over time, providing maximum flexibility. By combining the power of programmability with policy enforcement during the early stage of the SMTP transaction, spam bandwidth is eliminated, there is no quarantine folder to review, and the end user is empowered with complete control over their email privacy.

Binding policies to email addresses

A PEA is programmed via its policies and those policies may be assigned and/or changed at any time. Typical policies in today's version include:

- Pre-defined limits on the number of email messages accepted, e.g. "Allow the first 5 email addresses to come through and then allow no others".
- Scoping definitions to determine the count of either email addresses, domains, or subdomains, e.g. "Allow the first 5 domains to come through and then no others".

- Expiration of email operability on a date relative to either the date of creation *or* dynamically based on the date of the first time the PEA is used by a sender, e.g. "Do no accept any more email to this address 10 days after the first time this address is used."

Today's PEA policy also includes a *status*. Current status types include:

- **Enabled.** Accept email according to policy.
- **Disabled**. Do not accept any email to this address under any circumstances.
- **Locked**. Only accept email to this address if the sender has previously sent to this address. In this case, a sender's email address is compared during the SMTP transaction against a database of email addresses that have already sent to this PEA. If there is a match, the email is accepted. If there is no match, the email is rejected with a fatal error.
- **Validating**. Operate much like **locked** mode, and add challenge/response logic so that new senders may add their addresses to the database.

Both policy and status can be expanded in the future to incorporate other features which are detailed below in "Other Examples of PEAs".

Policy enforced at the edge

Today's email servers are fatally flawed in that, blacklisting notwithstanding, they *unconditionally accept properly addressed email* to all working addresses. At the SMTP transaction level, the following is a grossly simplified view of today's MTA logic:

1. Accept a connection
2. Accept the MAIL-FROM command
3. Accept the RCPT-TO command
4. Respond with fatal error if RCPT-TO does not exist, otherwise respond with OK.
5. Accept DATA
6. Queue the mail for delivery

The PEA implementation changes this logic slightly while maintaining full SMTP compatibility:

1. Accept a connection
2. Accept the MAIL-FROM command
3. Accept the RCPT-TO command
    a. Lookup the RCPT-TO and MAIL-FROM commands against the database of policies, statuses, and email addresses related to this specific RCPT-TO address.
    b. Execute the logic and determine if this sender is violating policy.
4. Respond with fatal error and if policy is violated.
5. Otherwise respond with OK.
6. Accept DATA
7. Queue the mail for delivery

It is the PEA's *conditional acceptance* of properly addressed email that is the core of its power. By making a novel use of the fatal error notification that already exists in SMTP, unauthorized email can be stopped *before it can be sent.* And because the error is fatal, it is nearly impossible for the sender to penetrate this defense and get email through to the recipient. PEAs give the end-user firewall-like control over their own personal email traffic.

Benefits of using Programmable Email Addresses

Policies can be changed and updated at any time

Each PEA contains a policy and a database of all email addresses sent from and received to that specific address. These policies can be updated and changed at any time. The decision to accept or reject email sent to a given PEA is determined by a combination of that PEA's specific policy and database *at the moment the SMTP transaction occurs.*

The ability to change policy at any time gives the end-user maximum flexibility to either loosen or tighten the restrictions controlling the inflow of email on a *per PEA basis.*

Saves bandwidth and infrastructure

PEAs stop unauthorized email at the earliest possible stage in the SMTP process and therefore its data cannot be sent. There are many significant benefits to this approach:

- Unwanted email does not require quarantine storage, eliminating disk storage costs.
- Bandwidth costs are eliminated as unwanted mail cannot be sent. It is important to note that all network hops in between the sending and receiving mail servers benefit from reduced bandwidth costs.
- There is no email content processing overhead related to filtering or analysis, reducing processing costs.

Unprecedented end-user empowerment

While typical content-based anti-spam technologies boasts 90%+ accuracy, they do little to help the end-user control unwanted email traffic that does not typically qualify has spam. PEAs go beyond content filtering and DEAs by empowering the end user in ways not possible before.

Unwanted email problem	PEA solution
Unwanted solicitations resulting from publishing an email address at a conference or other publicized events.	A PEA can be locked, allowing the initial legitimate senders to continue using the address while blocking new senders.
Receiving commercial email from marketing partners as a result of subscribing to a newsletter and inadvertently agreeing to redistribution of the subscriber's email address.	A PEA can be programmed to disallow redistribution, allowing the receipt of the newsletter but rejecting solicitations by other domains.
Email harassment from unwanted individuals.	That individual's email address can be marked for blacklisting which will then be bounced at the edge.
Email addresses gathered by a mailing list group are being distributed to spammers.	PEA policies can allow only the domains related to operating the mailing list alone and allow new legitimate users to validate themselves with challenge/response.
Subscription services that are technically unable to unsubscribe a user. For example, a user may have subscribed using an old email address that now forwards to their new address. The subscription process works by receiving an "unsubscribe" email and then unsubscribing the sender's email address. Because this sender's email address does not match the old subscribed address, unsubscribing becomes very difficult and time consuming.	A PEA can be disabled completely, disallowing any further email to that address.
Employee email relocation. When an employee leaves a company, their email is typically forwarded to their replacement. Unfortunately all personal publications and subscriptions are forwarded as well, creating an unnecessary burden for the replacement employee to deal with.	PEAs can be selectively disabled based on the new user's preferences.

Other Examples of PEAs

While today's PEA implementation targets spam and unwanted email, other future uses can create important functionality that simply is not possible with today's filtering or DEA technologies.

Parental email control

A "parental" PEA can be created for a given user's child and provide varying degrees of parental control over that email traffic, such as:

- Parental approval of all new email senders and/or recipients.
- Parental review/copy of each email sent/received.

Vacation email

Today's handling of email over a vacation period is a pure nightmare. On the return of a restful vacation, the user's inbox is filled with spam, newsletters that will not be read, error messages resulting from the vacation autoresponder sending to false addresses, and a small portion of legitimate email. A PEA containing a "vacation policy" can bounce all incoming email and send a vacation message only to legitimate email addresses. When the user returns from vacation they will have an empty inbox. Finally, a real vacation!

Business Hours

Certain email addresses may only want to receive email during business hours. Email received outside of business hours can be bounced and an explanatory email sent that outlines working business hours.

Dynamic Authenticity Binding

A PEA can be programmed to detect the degree of authenticity of the first email it receives, and then reject all subsequent email traffic that fails similar authentication tests. For example, a dynamically authenticating PEA might receive an email from an SPF friendly domain, therefore force all subsequent emails to be SPF-authenticated.

Incremental adoption of new email protocols

Because today's typical user has a single email address that handles all their incoming mail, any newly introduced email protocol must be 100% backward compatible to avoid loss of legitimate email. Because PEAs are programmable, they can adopt new email protocols that can only apply to a *small subset* of all the email received. This allows incremental introduction of new, richer, safer, and more secure email protocols without adversely affecting existing email traffic.

Comparions of PEA's vs. other technologies

Technology	How PEA's compare
Filtering	No false positives; no quarantine folders; nearly 100% effective in eliminating spam; zero bandwidth used for unwanted email; stops unwanted email this is not considered spam
DEA (e.g.	Has more than just an on/off state; continues working for certain

spamex, sneakemail, zoemail)	senders but not others; optionally employs challenge/response to add new authorized senders; can be programmed to accept a pre-determined number of different senders and disallow further redistribution; can change policy at any time; allows date-based expiration relative to first email address accepted.
C/R (e.g. Mailblocks)	No quarantine folders; no unnecessary challenges sent (for instance, an email sent to "support@" address is typically responded to by an individual who then receives a challenge); easy handling of newsletters and mailing lists.
Network (e.g. Barracuda)	All spam is stopped rather than merely throttled. Unwanted email that does not necessarily qualify as spam is stopped as well.

Features unique to the PEA

Programmable email addresses open up a new world of functionality that is outside today's filtering or DEA paradigm. A collection unique of features is below.

Prohibits unauthorized redistribution. Example: Certain newsletters use complex double-opt-in subscription processes whereby the opt-in confirmation email comes from one domain while subsequent newsletters come from another domain. A PEA can be programmed to allow the first 2 domains to pass only. Therefore the newsletter subscription and delivery processes flow seamlessly but any other sender is blocked from sending to that PEA.

Flexibly allows new authorized senders. A PEA can be set to "validate" new senders via challenge/response, allowing new authorized senders to add themselves to a PEA's database.

Set and forget operation. While the end-user can change a PEA's policy and database at any time, PEA's are programmed in advance to change their state based on various events. Our previous newsletter example that allows only the first 2 domains to pass will automatically block emails from other domains without any additional user intervention.

Dynamic Policies. Rules for determining emails can be dynamically created based on the characteristics of the incoming flow itself. In the simplest example we have an email address that expires *n* days after the first incoming email. More complex examples include determining the degree of authenticity on the first email received, (i.e., is it using SPF?) and then only allow subsequent emails that abide by the same standards.

Enables an entirely new set of email features. On/Off business hours, Parental Controls and Real Vacations are just a sample of some entirely new and useful features that can be implemented via PEAs

Enforces user policy at the gateway. PEA policies are applied to incoming email in the middle of an SMTP transaction and those that fail the test are bounced with a fatal error at the edge of an email network. For the first time, end-users have firewall-like control over their personal email traffic.

Adopt new email standards for a specific, user-defined email address. Because PEAs can change their policies at any time, end-users can apply new email standards against new or old addresses. Radical new email protocols can now be implemented without having to be concerned with backward compatibility because those protocols do not have to be universally applied.

Limitations

Users have to think about email in different way.

Today's email paradigm essentially gives an end-user a single address to use for all correspondence. With PEAs, users will need to acclimate to the concept of using different email addresses for different purposes. While there is flexibility in that an end-user can elect to use one email address to handle all ecommerce transactions or may decide to create a different PEA for each vendor, the fact remains that several email addresses will be in use at any moment.

Solution: The proliferation of free email services such as Yahoo and Hotmail has already given users the ability to create multiple email addresses. Many experts have advised users to stem spam flow by getting their own free email account that is disposable. The PEA concept formalizes this practice into an integrated user experience.

Users may get overwhelmed with too many email addresses to track

Over time, the user will have created many PEAs that are actively used, creating a management problem. Another issue might be Web sites that use the email address as the user id key, requiring end-users to track an email/password combination for certain sites.

Solution: This problem can be mitigated via improved user interfaces and tighter integration with the browser. Instead of requiring the user to manage their PEAs via today's "dashboard" interface, an extended browser would allow the user to create new and manage existing PEAs as part of the direct Web experience.

Does not address spam coming into email addresses like "support@", "sales@", "info@"

There are certain email addresses that are designed to allow all email to arrive. Certain job duties like salespeople or PR representatives do not want to limit their incoming email in any way. Essentially, these functions **do not** want email privacy and therefore PEAs will not provide an important benefit. In these cases a PEA infrastructure can still apply and allow a policy of "unconditional acceptance", passing all email to the user's inbox.

Conclusion

Programmable Email Addresses provide a new level of functionality that goes far beyond filtering or disposable email addresses. By allowing the end-user to assign policies to email addresses and using a relational database to track senders and policies, PEAs eliminate nearly all spam. Because users can set rules that are enforced at the network's edge, they finally complete control over their email privacy. PEAs can pave the way for entirely new email applications and protocols while maintaining compatibility with existing standards, making them the ideal bridge to sorely-needed structural improvements to SMTP.