Using DomainKeys Identified Mail (DKIM) with MDaemon

Alt-N Technologies

July 26, 2005

Abstract

DomainKeys Identified Mail (DKIM) is an open protocol for protecting email users against email address identity theft and email message content tampering. It does this by providing positive identification of the signer's identity along with an encrypted "hash" of the message content. Alt-N has taken a leadership position in the development and deployment of DKIM. MDaemon supports DKIM as well as Sender Policy Framework (SPF) and DomainKeys classic (DK).

Table of Contents

Email Security Problems and Solutions

Email Risks

Email is by far the most common service used on computer networks—local, wide-area and Internet. It provides easy, quick, low-cost and reliable communications for personal, educational and business purposes.

Because of its openness, simplicity, and worldwide deployment, email is also easily and often abused. Unsavory individuals, groups and businesses misuse email to deceive, steal, and destroy through:

- **Spoofing** — the unauthorized use of an email address or domain.

- **Spam** — the sending of unsolicited/unwanted email.

- **Viruses/Trojans** — content designed to harm or provide unauthorized information and/or access to a computer system.

- **Phishing** — the attempt to steal sensitive information—IDs, passwords, credit card numbers—through email and the web using spoofing.

Email address spoofing is a fundamental risk to legitimate email users. Using false addresses enables the spread of other types of risks. The developers of spam, viruses and phishing schemes largely depend upon their ability to easily falsify senders' addresses.

Email User Confidence

For email to continue to be beneficial, the impact of email risks must be reduced and, hopefully, eliminated. Message senders must have confidence that no one is spoofing their email addresses and that each message they send arrives unaltered. At the same time, recipients must be assured that each email is from the claimed sender and that the received messages are the same as the ones sent. Stated another way, beyond ease-of-use, speed and low cost, the requirements of email users are:

- Assurance of sender identities.

- Protection against message tampering.

While a third need—message encryption—has application for high-security email, for most users, content ciphering is an overly complex waste of time and resources. Filling the primary needs of email users requires authentication for both senders and content.

Note: Assuring the identity of a sender does not tell you much about the reputation of that sender. While identity assurance alone is not enough for complete security, it does provide a basis for establishing reputations. Without an assurance of who is sending you a message it is impossible to gauge the reputation of that sender.

Authentication Methods for Individuals

Authentication technology is nothing new to email, but its application has largely been limited to individuals requiring security in most, if not all, messages. Such security measures are usually deployed on personal email clients—not on email servers—and require the active participation of both senders and recipients. With PGP and similar ciphering technologies, for example, senders use private keys to encrypt or sign messages or do both. Recipients validate the sender and the message content by using the counterpart public keys, available from the sender or from a public key server. All signing and validation is done on the client machines, and requires periodic updating of the private keys for senders and the public keys for recipients. Sometimes, third-party certificate authorities are also involved for creating and authenticating the key certificates. These approaches can be tedious to maintain.

For the typical email domain, authentication methods are more quickly and easily deployed at the mail server level, using the domain name server infrastructure for accessing the public keys.

Authentication Methods for Email Servers

Email authentication can be broadly divided into two camps:

- *path authentication* techniques—such as SPF and Sender-ID—which verify that a message is being sent from a source authorized to do so.

- *cryptographic authentication* techniques—such as DomainKeys and IIM—which verify the actual message itself, not just it's delivery path, by inserting a digital signature created using a key known only to the message signer.

In the cryptographic authentication space, although the details differ, sometimes drastically, the basic approach is similar:

- The sending entity establishes a private key they never share and a public key which they publish either in a domain name server record or some other public access facility.

- Before sending an email, the sending entity calculates and adds a message header containing a digital signature or "fingerprint" of the message using their private key.

- Upon receiving a message, the receiving server validates the signature using the sending entities' public key which it retrieves from the domain name server or other public facility.

- Based upon whether the signature verifies, fails to verify, or is missing completely, the recipient server can apply appropriate local policy to the message.

In cooperation, multiple organizations have joined together and taken the first steps toward industry-wide standardization of cryptographic authentication. The product of this join effort is called DomainKeys Identified Mail (DKIM).

DomainKeys Identified Mail

DKIM Background

DomainKeys Identified Mail (DKIM) combines the primary functions of two authenticating technologies—DomainKeys (DK) by Yahoo! and Identified Internet Mail (IIM) by Cisco Systems. The effort to create DKIM has brought together these two organizations, plus other participants such as Alt-N Technologies, America Online, Brandenburg Internetworking, EarthLink, IBM, Microsoft, PGP Corporation, Sendmail, StrongMail Systems, Tumbleweed and VeriSign.

DKIM has been designed for quick, easy and low-cost implementation. It has these basic features:

- Message signatures are placed in message headers to avoid the confusion often created when signatures are part of the body text.

- Email server operators can create their own public/private key pairs, providing independence from the cost of third-party trusted certificate authorities creating the key pairs.

- Transparency to and compatibility with the existing email infrastructure

- No new infrastructure requirements.

- Server implementation to reduce deployment time. No changes required of clients.
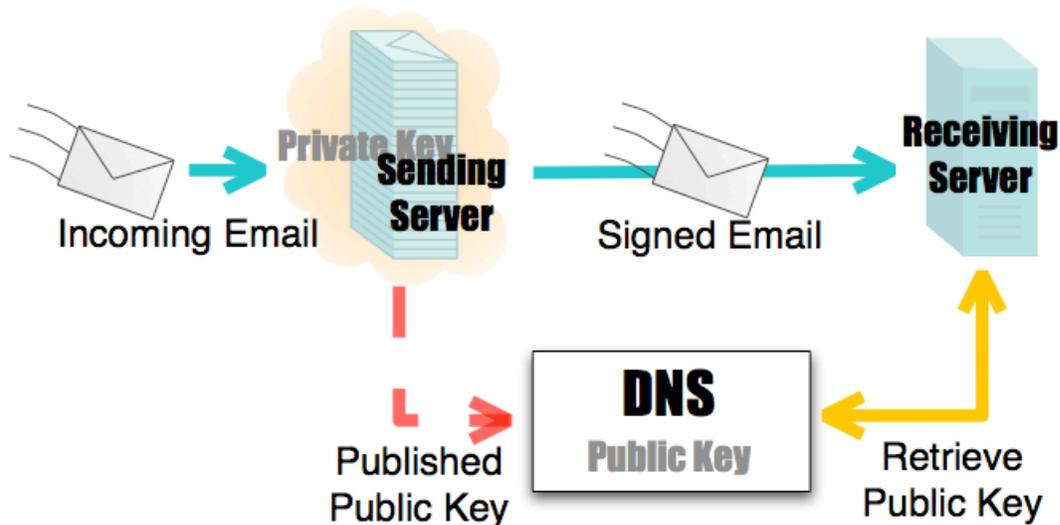
- Incremental deployment.

DKIM Technology

DKIM is an open protocol for assisting in the detection of forged email messages and email content tampering. It does this by providing positive identification of the signer's identity and securing an encrypted "hash" of the message content.

An important feature of DKIM is its compatibility with the existing DomainKeys infrastructure. DKIM utilizes existing DomainKeys key records allowing it to leverage the existing and already widely installed base of DomainKeys users and software. This includes users of Yahoo! Mail, Earthlink, Google's GMail, and the users of MDaemon 8 which has included DomainKeys support for some time.

DKIM uses public key/private key technology to sign and authenticate messages. A private key resides in a protected but internally accessible part of the signer's network. The

corresponding public key is stored in the signer's domain name server or other publicly accessible facility. The use of "selectors" allows each domain to have multiple key pairs. Selectors are labels for enabling multiple private/public key pairs within one domain. Selectors might identify different servers within a domain or different secondary domains within a server, for example.

Selector names can be almost any combination of letters and numbers, such as london, chicago986 and 2005_acc in these examples of DKIM selectors:

london._domainkey.poboxes.shacknet.nu

chicago986._domainkey.poboxes.shacknet.nu

2005_acc._domainkey.poboxes.shacknet.nu

To configure and use DKIM:

1. The system administrator creates a private/public key pair for the server and publishes the public key in the domain's domain name server.

2. Using the private key, the sending server creates a signature for each outgoing message. The resulting signature data is stored in a "DKIM-Signature" header within the message.

3. The receiving server obtains the signature from the "DKIM-Signature" header and verifies it using the signer's public key.
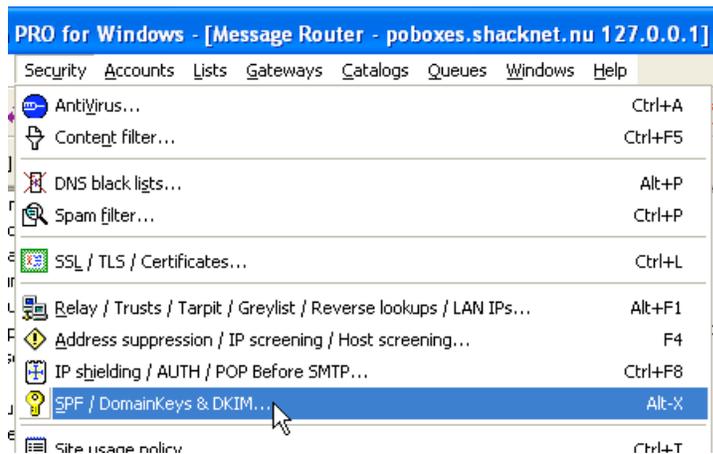
Authentication success occurs if the message signature "verifies," meaning it is successfully tested against unaltered message content using the public key from the signing domain. Authentication failure occurs when a signature does not verify or when a signature is missing from a message claiming to be from a domain which always signs messages. When either occurs, the receiving server has an additional basis upon which to apply local policy.

Using DKIM with MDaemon

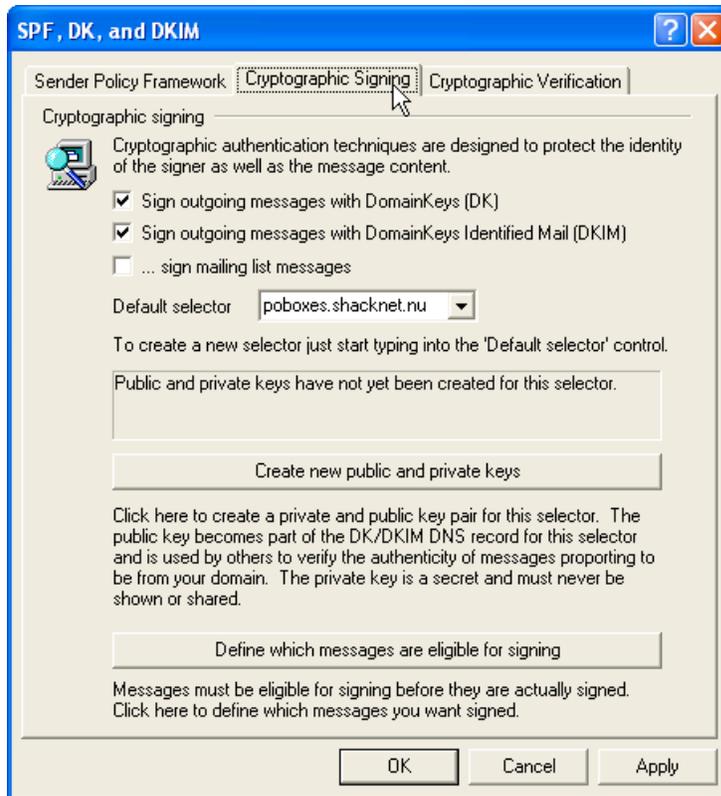MDaemon supports DomainKeys Identified Mail (DKIM), plus DomainKeys for backward compatibility.

DKIM Access

Access to DKIM setup is through the SPF / DomainKeys & DKIM command in the Security menu.

Cryptographic Signing Tab

The Cryptographic Signing tab contains options for setting up outgoing message signatures.

These settings determine if you are using DomainKeys or DomainKeys Identified Mail or both. Signing can optionally apply to mailing lists. If messages to mailing lists are included in the DKIM processing, MDaemon signs each outgoing message to each list member.

Creating Public/Private Keys

Using DKIM requires creating at least one public/private key pair.

Creating public/private key pairs with MDaemon involves entering a selector name and pressing the button named "Create new public and private keys". MDaemon automatically generates a default selector called "MDaemon" and creates a public/private key for use with this selector.

Creating a selector and associated key pair displays example information to enter into the DNS record for your domain.

Specify Signing Addresses

MDaemon signs messages based on their source or designation addresses.

Specifying which messages to sign involves pressing the "Define which messages are eligible for signing" button. Doing so opens a file which allows you to enter the addresses.
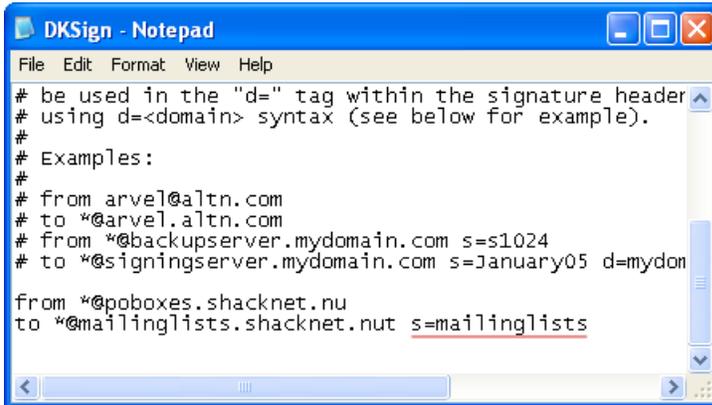

By default, MDaemon makes all messages from users of your primary domain eligible for signing.
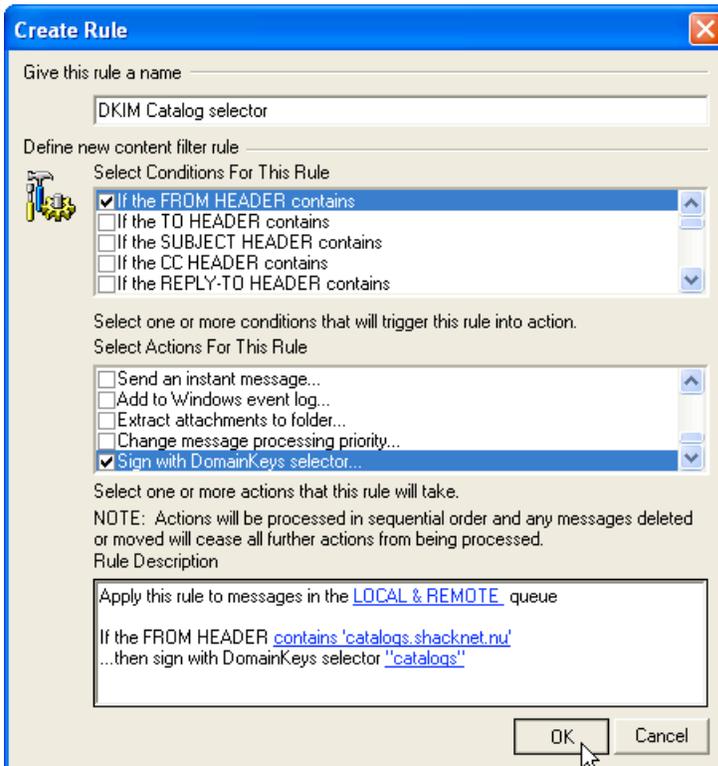
Using DKIM Selectors

MDaemon uses the default selector to sign all messages. However, you can configure the Signing Address file or the Content Filter to have MDaemon to use different.

In the address file you can specify a selector for each address.
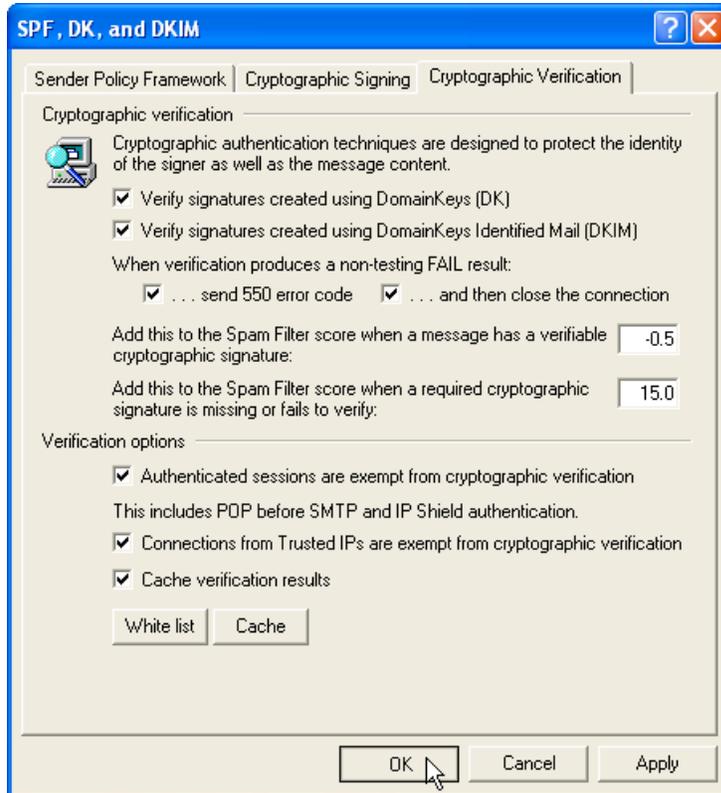

The Content Filter has options specifically designed for processing DKIM selectors.

This provides flexibility in implementing DKIM within MDaemon.

Cryptographic Verification Tab

The Cryptographic Signing tab contains options for setting up message authentication.

These settings determine if you are authenticating DomainKeys or DomainKeys Identified Mail or both. These settings also determine the policies for handling messages failing the DKIM authentication tests. Options vary from doing nothing to closing the SMTP session. White lists are available for exempting specified domains from authentication.

Summary

DKIM is an emerging industry-wide proposal for authenticating email identities and message content. MDaemon offers an easy to use and complete implementation of DKIM for the Windows platform.