eMill

PROFESSIONAL EMAILING SOLUTION

WHITE PAPER

THE SENDER AUTHENTICATION METHODS

Authentication methods are the most used and implemented techniques to detect Spam. It is based on the fact that spammers always try to hide who they are and where they come from. Therefore, if you are not able to identify a sender, it should mean that the message is a Spam.

The impact for email marketers is that you should be able to answer correctly to a server when it asks “Who is sending this email?”.

The implementation of authentication methods will allow you to give the correct answer. It mainly consists in associating the IP address of the computer you use to send emails to a known domain name which authorize this computer to send emails.

I. Definitions

Before starting to detail how the authentication methods work and how they can be implemented, it is necessary to define the following key words:

SMTP Server

The Simple Mail Transfer Protocol (SMTP) is an internet standard protocol for sending electronic mail messages between computers. It establishes the connection between the sender and the receiver. The receiving server is usually the one which tries to authenticate the email sender.

IP Address

An IP address is composed of 4 group of figures between 0 and 255 (ie. 192.168.78.90). It aims at identifying each computer connected on Internet. Residential addresses are most of the time dynamic. This means that the IP address of your computer is changing regularly. In the opposite, companies usually get a fixed IP address.

Domain name

A domain name is the ‘literary’ equivalent to the IP address. It aims at identifying a computer more easily than with group of figures (ie. the eMill domain name is eMill.net). Note that the domain name of a web site corresponds to the IP address of the computer where the site is hosted.

DNS Server

A DNS server is a software that makes the conversion between IP addresses and domain names.

For instance, when you enter the address http://www.emill.net in your web browser, a set of DNS servers are consulted to know what is the corresponding IP address.

A DNS server includes several fields:

- **The A field** which indicates an IP address. It aims, for instance, at providing an IP address when the DNS server is consulted about the ‘emill.net’ domain name.
- **The CNAME field** which indicates the domain name (alias).

- **The MX field** which indicates the domain name of the SMTP server(s) authorized to send and receive emails.

- **The NS field** which indicates the address of the primary and secondary DNS server.

- **The TXT fields** used to implement specific techniques.

The DNS server must be properly configured before sending emails but also for any communication on the local or remote networks. Consequently, it is most of the time correctly configured. However, if you have any doubt, you can enter your company domain name at http://www.dnsreport.com/ (in the DNS Report field). If one of the parameters is displayed in red, you should consult your network administrator.

DNS resolve & reverse DNS

When you enter a web site address (ie. http://www.emill.net) in a web browser, DNS servers will **communicate the IP address corresponding to this domain**. This process is called DNS resolve.

The **reverse DNS process consists in consulting a DNS server to get the domain name corresponding to an IP address**. This process is the authentication method the most commonly used. For instance, the AOL servers, which use this technique, retrieve the IP address of the sender which is located in the email header. Then, they ask to DNS servers if a domain name corresponds to this IP address. If they receive a negative answer they don't deliver the message.

II. The Sender Policy Framework

SPF is an **open standard** which sets an authentication method based on the email sender domain name.

In order to understand how it works, it is important to know that 2 sender email addresses are used when sending an email (these 2 addresses are most of the time the same):

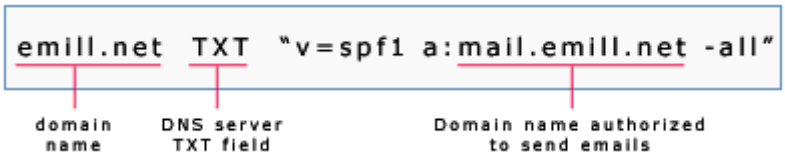
- The '**Mail From**' is the email address used by SMTP servers to communicate. It corresponds to the email envelope.
- The '**From**' is the email address which is displayed to the receiver within its mail client. It corresponds to the email header.

SPF is a technique based on the communication between SMTP servers. Therefore, the 'Mail From' address will be used to identify the sender.

The sender is authenticated in two steps:

- **The receiving server retrieves the sender domain name** indicated in the 'Mail From' field of the email header. For instance, if the email address is philippe@mail.emilltest.com the receiving server will use the domain 'mail.emilltest.com'.
- **The receiving server consults the DSN servers** to know if this domain name is authorized to send emails.

In order to answer correctly to the server, you need to add a setting in your DNS server which tells which machines are authorized to send emails. This parameter has the following format:



To create a record like the one above, the SPF community offers an online wizard at: http://www.openspf.org/wizard.html.

III. Sender ID

Sender ID is a standard protocol combining the **Sender Policy Framework** (see above) and **CallerID** created by Microsoft. It aims at identifying the email sender domain name.

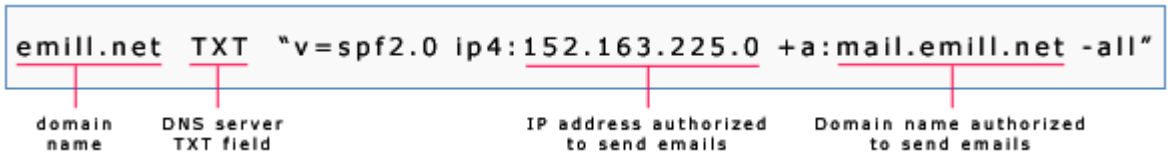
Acknowledged by the IETF (Internet Engineering Task Force) which establishes Internet standards, Sender ID is one the authentication methods the most used: AOL, Gmail, Live, B2B… Among the main webmails, Yahoo is the only one which does not support Sender ID. Certainly because they released their own authentication method, DomainKeys (see part 4).

Sender ID is based on the same idea as SPF:

- **The receiving server retrieves the sender domain name** in the email header. Comparing to the SPF, Sender ID does not only look for the domain name in the 'Mail From' field but browse other e-mail headers.
- **The receiving server asks the DNS server** to know if the sender IP address or domain name is authorized to send emails.

In order to answer correctly to the receiving server request, a specific TXT field must be added to the DSN server settings. To implement Sender ID, you must follow 4 steps:

- **Define the IP addresses of the machines which send emails on your network.** If a 3rd party company is sending emails on your behalf (ESP, hosting company), you must indicate their domain name (and not their IP addresses if they have implemented Sender ID).
- **Create the Sender ID record** using the wizard provided by Microsoft: http://www.microsoft.com/mscorp/safety/content/technologies/senderid/wizard/
- **Publish the Sender ID record** on your DNS. This record has the following format:



- **Test the Sender ID implementation** from the following website: http://senderid.espcoalition.org/.

You will find below complementary resources available online in order to learn more about Sender ID and its implementation:

- Website dedicated to Sender ID: http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx
- RFC standard established by the IETF (for advanced users only): http://www.ietf.org/rfc/rfc4406.txt?number=4406

IV. DomainKeys

DomainKeys is a norm created by **Yahoo!** that aims at identifying the sender domain name but also to be sure that the email has not been modified during its transfer. As DomainKeys follow 2 objectives, it is more complex to implement and it needs more resources. That is the reason why only **Yahoo and Gmail** support this protocol.

Note that Yahoo! currently works with other companies (AOL, EarthLink, Microsoft, Sendmail, StrongMail…) in order to **standardize this protocol** under the name of DomainKeys Identified Mail (DKIM).

Here is how **DomainKeys** is running its authentication process:

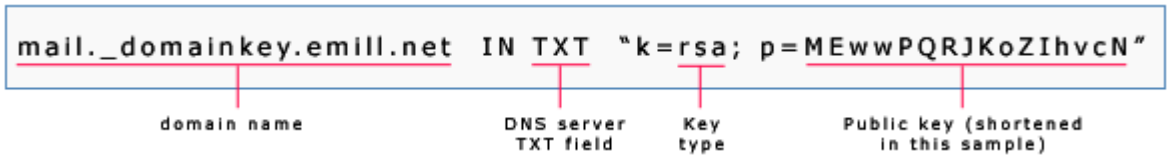
- The domain owner renders **a public and a private key.**
- The public key is added to the DNS server and the private keys are stored on the machines that send emails on the network.
- When sending an email, the sending server uses the private key to digitally sign the message. The signature is added to the email header.
- The email is transferred to the receiving server corresponding to the message recipient.
- The receiving server retrieves the email signature and the sender domain name indicated in the 'From' field.
- The receiving server contacts the sender domain DNS server and checks if its public key corresponds to the email private key.

This process allows you to authenticate the sender domain name and, thanks to the key, you are sure that the email header and content have not been modified.

In order to implement this authentication method, you must first **get a mail server which supports DomainKeys** in order to add the private key. Among these mail servers, you will find Sendmail, Qmail, Exchange 2003, Port 25, Etype, XMServer, Ecelerity, StrongMail, MDaemon, Postfix, IronPort, L-Soft… Consult your server documentation to know if it is compatible with DomainKeys.

Then, you can **start the implementation of DomainKeys**:

- **Render the keys**: Most of the mail servers quoted above have a tool to render the private and public keys. You can also consult the following website to render the keys by yourself: http://domainkeys.sourceforge.net/keygen.html.
- **Add to the DNS Server a TXT field with the public key.** The parameter should get the following format:



- **Add to the mail server(s) the private key(s)** (check your mail server documentation).

You will find below some complementary resources available online in order to learn more about DomainKeys and its implementation:

- Website dedicated to DomainKeys: http://antispam.yahoo.com/domainkeys
- Implementation wizard for several mail servers: http://domainkeys.sourceforge.net/

eMill

PROFESSIONAL EMAILING SOLUTION

Active+ Software
51, Avenue Général de Gaulle - 66320 – Vinça – France
Phone: +33 4 6805 4774 – Fax: +33 4 6805 5701
E-mail : info@activeplus.com