Configurare Filtri AntiSpam in Postfix

In un precedente articolo abbiamo descritto la struttura generale di un servizio di posta elettronica, lato server. In questo articolo ci concentriamo su di un solo elemento del sistema globale, il server anti-spam. Per essere fedeli al titolo di questa rubrica, daremo delle indicazioni su come configurare Postfix (versione 1.x), uno tra gli MTA più apprezzati dagli specialisti per le sue prestazioni, la sua scalabilità e la sua provata sicurezza.

Visto lo spazio a disposizione, ci limiteremo a descrivere alcune semplici configurazioni, indicando poi ulteriori possibilità.

Per semplicità assumiamo di aver già configurato correttamente Postfix come server SMTP che accetta i messaggi di posta da internet e li invia al successivo server anti-virus.[1] In particolare il *relay-smtp* deve essere permesso solo verso il server anti-virus e solo verso i domini di posta della nostra organizzazione.

La parte principale della configurazione anti-spam in Postfix, usualmente nel file */etc/postfix/main.cf*, è nella direttiva `smtpd_recipient_restrictions`. Questa non è l'unica direttiva che accetta regole anti-spam, ma è di solito la più conveniente in quanto le regole da lei specificate vengono applicate dopo lo scambio `HELO`, `MAIL FROM:` e `RCPT TO:` .[2] I vantaggi sono:

1. si conosce l'indirizzo del destinatario di cui resta traccia nel file log;
2. si possono definire destinatari che ricevono tutto lo spam;
3. esistono clienti, in particolare alcuni che inviano spam, che 'impazziscono' se si invia un messaggio di errore prima di `RCPT TO:` e riprovano immediatamente.

La direttiva `smtpd_recipient_restrictions` può essere configurata come segue (l'ordine dei parametri può essere modificato secondo le esigenze locali):

1Si veda *Progettare un'Architettura di Posta Elettronica* in ICT Security 15, Settembre 2003.

2 Si veda ad esempio *Network Troubleshooting 101* in ICT Security 16, Ottobre 2003 o una qualunque referenza sul protocollo `SMTP`.

```
smtpd_recipient_restrictions    =    reject_non_fqdn_sender,
      reject_unknown_sender_domain,
      permit_mynetworks,
      reject_unauth_destination,
      check_sender_access hash:$config_directory/trusted,
      check_recipient_access pcre:$config_directory/recipient-pcre,
      check_recipient_access hash:$config_directory/spammed,
      reject_unauth_pipelining,
      reject_invalid_hostname,
      check_helo_access pcre:$config_directory/helo-pcre,
      check_sender_access hash:$config_directory/access,
      check_client_access hash:$config_directory/access,
      check_client_access hash:$config_directory/no-rbl,
      check_client_access rbl:$config_directory/dnsbl-filter
```

`reject_non_fqdn_sender` e `reject_unknown_sender_domain` bloccano messaggi con mittenti il cui dominio non è nella forma FQDN (Fully Qualified Domain Name) e non ha un record A o MX nel DNS (Domain Name System). `permit_mynetworks` e `reject_unauth_destination` permettono solo il relay verso (o da) la nostra rete interna.[3] Il parametro `check_sender_address` dice di controllare se il mittente è indicato nel file *trusted* nella directory di configurazione in modo che la posta dai mittenti indicati sia accettata incondizionatamente. I due parametri `check_recipient_access` filtrano il primo tramite delle *regular expression* indirizzi di destinazione impossibili, il secondo invece elenca degli indirizzi che accettano tutta la posta, spam incluso. `reject_unauth_pipelining` e `reject_invalid_hostname` bloccano messaggi che violano il protocollo SMTP, mentre è pericoloso adottare `reject_unknown_hostname`, `reject_non_fqdn_hostname` e `reject_unknown_client` che bloccano molta posta valida inviata da client o server che specificano il proprio hostname nella stringa di `HELO` non in formato FQDN, o senza record DNS A o MX valido, o senza risoluzione DNS inversa. `check_helo_access` filtra indirizzi indicati nella stringa di `HELO` tramite le *regular expression* specificate nel file *helo-pcre*, ottimi candidati sono: `/^\$/`, `/^%/`, `/^#/`, `/^propriodominio$/`, `/^pro.pr.io.IP$/`, `/^test\d*$/`, `/^mail.com$/`, `/^hellrimore\d+\.com$/` ecc.
La lista specificata nel file *access* contiene un elenco di indirizzi e domini di posta da bloccare, sia come mittenti che come destinatari, perché noti spammer o invalidi. Questa black-list è gestita manualmente ed aggiornata quando necessario.

Nel file *no-rbl* vi è una lista di mittenti che non vogliamo siano bloccati anche se per caso capitassero in una lista DNSBL (DNS-based Blocking List). Infine nel file *dnsbl-filter* mettiamo l'indirizzo delle liste DNSBL, ovvero liste distribuite via DNS in tempo reale che indicano indirizzi

3 E' fondamentale che vi sia il parametro `reject_unauth_destination` per chiudere il relay a terzi.

mittenti di posta elettronica noti per essere sorgenti di spam.

Vi è grande discussione e pareri discordanti sull'uso delle black-list DNSBL, ma non è questa la sede per addentrarci nell'argomento. Piuttosto possiamo segnalare che esistono liste di tipo diverso, che contengono elenchi di proxy, di relay aperti, di sorgenti dirette, di formmail, di dialup e IP dinamici. Un elenco parziale di liste DNSBL attualmente in uso[4] è: SBL (una delle più prestigiose e storiche), DSBL, CBL, PSS, OPM, SPEWS, Easynet Blackholes, NJABL, ORDB (classica per i relay aperti), Easynet Dynablock, PDL. Di norma si utilizzano da 1 a 3 liste, a seconda delle necessità.

In ogni caso quando si rifiuta un messaggio via una DNSBL è opportuno che il messaggio di errore inviato al mittente indichi il motivo del blocco (sommariamente), e fornisca una indicazione di contatto (una URL e/o un indirizzo non protetto dai filtri DNSBL). Questo è soprattutto vero nel caso di IP dinamici, poiché sono spesso bloccati completamente in quanto sovente utilizzati per l'invio di spam o virus tramite proxy aperti, worms e in genere macchine infette di utenti finali. In questi casi si può ovviare al blocco inserendo il legittimo dominio o indirizzo email mittente nel file *no-rbl*.

Se il messaggio in arrivo supera tutti i test della direttiva `smtpd_recipient_restrictions`, allora il mittente da il comando DATA ed invia il messaggio. Il messaggio è composto da due parti, gli *header* ed il *body* (ricordiamo che i test elencati sino ad ora sono fatti utilizzando le informazioni disponibili a livello di *envelope smtp*, ovvero esterne al messaggio stesso). Possiamo di nuovo porre dei filtri sugli *header* del messaggio utilizzando la direttiva `header_checks` con delle *regular expression*:

```
header_checks = pcre:$config_directory/header_checks
```

Un esempio del file *header_checks* è in Tabella 1.

Analoghi filtri possono essere fatti anche sul *body* del messaggio utilizzando la direttiva `body_checks`. Va notato subito però che questi controlli sono molto più pesanti, visto che la quantità di dati da analizzare è molto superiore, e pertanto posso rallentare molto il flusso della posta. Inoltre questi controlli sono molto simili a quelli

4 Questi servizi sono molto dinamici e cambiano velocemente.

dell'anti-virus e forse sarebbero meglio implementati in quella sede. Vi sono comunque vari programmi specializzati nel *Content Filtering*, quali SpamAssassin, che possono essere invocati per analizzare il body dei messaggi. Bisogna infatti ricordare che per avere filtri efficaci bisogna espandere anche tutti gli attachment, come fanno gli anti-virus, interpretando i vari tipi di file appesi al messaggio, e questa analisi richiede ovviamente tempi e tecniche diverse da quelle descritte in questo articolo. In ogni caso la direttiva `body_checks` può essere utilizzata per controlli d'emergenza e limitati.

Andrea Pasquinucci
Consulente di Sicurezza Informatica
pasquinucci@ucci.it

Furio Ercolessi
Coordinatore dei Servizi di Messaggistica di Spin Srl
furio@spin.it

Riferimenti Bibliografici:

[1] Informazioni su spam: http://www.collinelli.net/antispam/, http://spam.abuse.net/, http://www.mail-abuse.org/, http://www.spamhaus.org/ ed altre risorse on-line

[2] http://www.postfix.org

[3] Per la descrizione dei parametri accettati da `smtpd_recipient_restrictions` si veda il file *sample-smtpd.cf* nella documentazione di Postfix.

```
/.*\@public\.com/                                  REJECT
/^X-Originating-IP:.*(\[|\(|\s)80\.248\.64\.50/  REJECT Source of 419 scam.
/^From:.*himailer\.com/                            REJECT Himailer can get lost.
/^X-Mailer: Advanced Mass Sender/                  REJECT
/^Message-ID: <0000[0-9a-f]{8}\$0000[0-9a-f]{4}\$0000[0-9a-f]{4}\@/    REJECT
/^Subject: \=\?euc\-kr\?B\?W7GksO1d/               REJECT
```

Tabella 1: Esempio del file *header_checks*