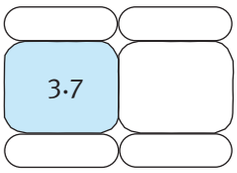
PERCHÉ È COSÌ DIFFICILE COMBATTERE LO SPAM?

Ernesto Damiani

L'invio di messaggi di posta indesiderati, il cosiddetto *spamming*, sta raggiungendo livelli preoccupanti e le previsioni per il futuro sono ben poco rassicuranti. Com'è possibile che un'intera comunità di esperti composta da aziende, utenti individuali, produttori di software e ricercatori informatici sia tenuta in scacco da un ristretto numero di malintenzionati? In questo articolo si analizzano i motivi tecnologici e organizzativi che rendono difficile combattere lo *spam*, descrivendo gli strumenti software utilizzati dagli *spammer* e quelli a disposizione di chi combatte lo *spam*.



1. INTRODUZIONE

Tra le tecnologie di Internet, la posta elettronica è forse quella che ha più radicalmente cambiato il modo di vivere e di lavorare di centinaia di milioni di persone in tutto il mondo, incluso chi scrive e quasi certamente anche chi sta leggendo quest'articolo. Purtroppo, però, l'utilizzo e la gestione del servizio di posta sono resi sempre più disagevoli dalla marea di messaggi non voluti, collettivamente noti come *spam*[1] che ognuno di noi riceve ogni giorno. Le tecniche di difesa *antispam* sono state oggetto di intenso lavoro negli ultimi cinque anni, ma purtroppo la tecnologia per l'invio di *spam* è migliorata almeno altrettanto.

All'inizio lo *spam* consisteva soprattutto di singoli messaggi inviati attraverso server SMTP compiacenti o mal configurati. Oggi si tratta quasi esclusivamente di messaggi generati dinamicamente e inviati su vasta scala attraverso strumenti software concepiti appositamente. In questo articolo mettiamo a fuoco la natura del problema dal punto di vista tecnologico (il lettore interessato agli aspetti legali può consultare l'**Appendice 1** a p. 51) presentando l'evoluzione degli strumenti per l'attacco (gli strumenti "*malware*" per l'invio di *spam*) e le principali tecniche di difesa (i filtri) oggi disponibili.

[1] Il termine *spam* viene dal nome di un cibo in scatola considerato poco appetitoso e dal sapore piatto, ben poco attraente soprattutto se – come i messaggi indesiderati – viene servito sempre, a pranzo e a cena. Il tormentone *spam, spam, spam* per indicare qualcosa di noioso e ripetitivo è stato usato, tra gli altri, dal noto gruppo comico dei Monty Python nel loro storico show "Monty Python's Flying Circus", ambientato in un locale dove ogni pietanza proposta dalla cameriera era a base di *spam*.

2. LA TECNOLOGIA DELLA POSTA ELETTRONICA

La nascita della posta elettronica risale al 1972, quando Ray Tomlinson installò su ARPANET un sistema in grado di scambiare messaggi tra le varie università connesse alla rete, ma chi realmente ne definì poi il funzionamento fu John Postel. Tutta la posta elettronica spedita su Inter-

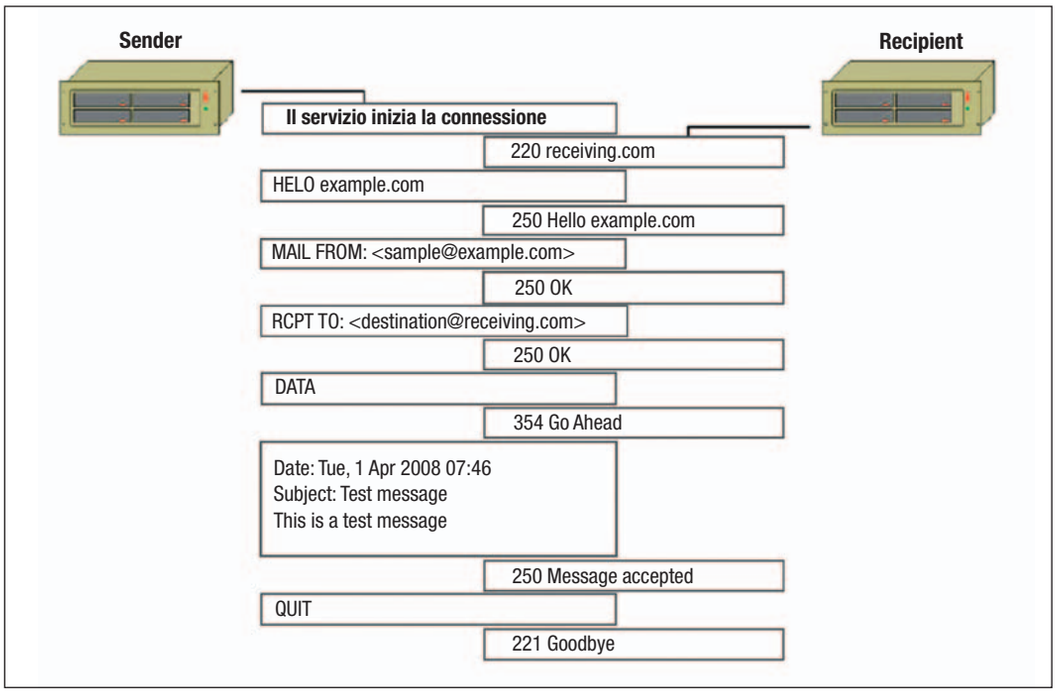


FIGURA 1
Un recapito SMTP

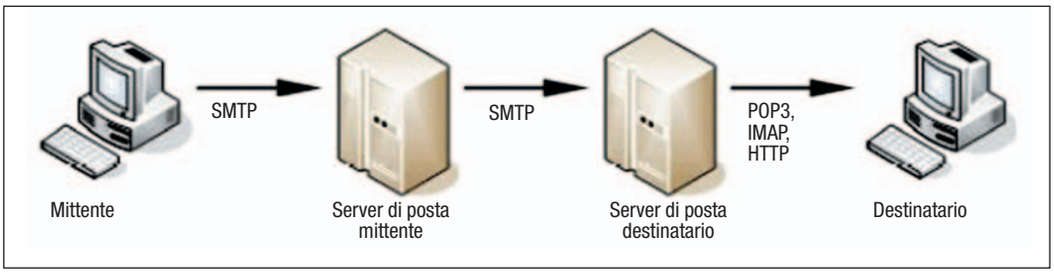


FIGURA 2
La consegna SMTP

net viene trasferita usando un unico protocollo: lo *Standard Mail Transport Protocol* (SMTP), definito da Postel nella RFC 8219 e implementato in centinaia di strumenti software (setti spesso *Mail Transfer Agent* o MTA) come il ben noto `sendmail`.
Si tratta di una tecnologia standard: ogni server Internet che utilizza SMTP è in grado di inviare e ricevere posta da qualsiasi altro server SMTP su Internet. Per capire come funziona SMTP basta dare un'occhiata alla figura 1, che mostra uno scambio di messaggi tra un MTA mittente che ha un messaggio di posta da trasmettere e un MTA ricevente che accetta il messaggio perché diretto a un indirizzo di posta da lui gestito. Inizialmente viene aperta una sessione sulla porta TCP 25, e segue una serie di messaggi, in alternanza tra client e server, che iniziano tutti con un codice numerico di tre cifre.
Ogni messaggio di posta è diviso in un'intestazione, composta dei campi `Date:` e `Subject:` mostrati nella figura (e da altri campi come `From:` che contiene l'indirizzo del mittente, `To:` che contiene l'indirizzo del destinatario e `Return-Path:` che contiene l'indirizzo da usare per la risposta) e dal corpo del messaggio, che contiene il testo vero e proprio.
Va notato che tutti questi campi fanno parte del blocco dati del messaggio, per il quale SMTP non prevede alcun meccanismo di verifica o controllo. Naturalmente il passaggio della posta tra i due server SMTP non esaurisce il percorso di consegna del messaggio il server mittente ha sicuramente ricevuto il messaggio da un client, e il ricevente lo consegnerà probabilmente a un altro client (Figura 2), il vero destinatario finale, attraverso appositi protocolli di consegna come IMAP e POP, su cui non ci soffermeremo qui[2].

[2] I messaggi di posta possono anche essere recapitati dopo essere stati incapsulati in altri protocolli applicativi, quali HTTP (*HyperText Transfer Protocol*), come avviene nei sempre più diffusi servizi di Webmail.

Il protocollo SMTP è uno dei più vecchi protocolli di Internet ed stato volutamente mantenuto semplice, visto che un server SMTP deve poter gestire decine di connessioni al secondo. Questa semplicità si traduce però in vulnerabilità, perché le due informazioni identificative che il server mittente passa al destinatario (il proprio nome e l'indirizzo e-mail a cui il messaggio è diretto) non vengono verificate da quest'ultimo e possono essere quindi facilmente falsificate.
Per chiarire questo punto, esaminiamo meglio un campo Received: facente parte dell'intestazione di un messaggio di posta:

```
from 159.149.70.1 by pollon (envelope-from <caio@crema.unimi.it>,
uid 201) 08 Dec 2008 18:42:20 -0000
```

Questo campo dice che il messaggio è stato ricevuto dal server SMTP che si chiama pollon (come dice la clausola `by pollon`) e proviene da un MTA di cui non è noto il nome, ma che ha l'indirizzo IP 159.149.70.1. Osserviamo subito il campo `envelope-from`, che non è il campo `From`: all'intermo del messaggio, ma quello che fa parte dell'intestazione SMTP. Contiene l'indirizzo (`caio@crema.unimi.it`) e la user-id (201) del mittente sul MTA di provenienza. Una precauzione che utilizzano molti MTA "diffidenti" è rifiutare posta elettronica in cui il contenuto del campo `envelope-from` dopo la chiocciola non è traducibile dal DNS (il servizio di traduzione nomi-indirizzi di Internet), cioè non è un *Fully Qualified Domain Name* (FQDN) ma solo un frammento non traducibile come caio@crema[3].
Questo può accadere quando l'editor di posta elettronica usato dall'utente (esempio, Outlook o Eudora) genera lui stesso i campi SMTP invece di lasciarlo fare al MTA, ma è anche un indizio che chi manda il messaggio potrebbe avere qualcosa da nascondere. Oltre a questo, vi sono altri due elementi importanti da osservare:

- possiamo ritenere l'intero campo `Received`: affidabile solo se conosciamo e consideriamo fidato il server `pollon` che l'ha creato. Altrimenti la riga potrebbe essere falsa;
- se il campo `Received`: è considerato affidabile, la parte importante è `from` 159.149.70.1. Di questo indirizzo IP ci fidiamo, perché l'ha controllato il nostro server fidato pollon quando ha ricevuto il messaggio. Anche qui, come per `envelope-from`, si può usare il DNS per un controllo; ma in questo caso si tratta di una query DNS inversa per ricavare dall'indirizzo il nome del server che ha consegnato il messaggio a `pollon`, e poi usare il comando `whois` per conoscere la persona e l'organizzazione a cui l'indirizzo è stato associato. Ecco, in sintesi, il risultato di `whois` per questo messaggio:

```
# ARIN WHOIS database, last updated
2008-12-08 19:10
# Enter ? for additional hints on
searching ARIN's WHOIS database
% Information related to
 '159.149.0.0 - 159.149.255.255'
inetnum:    159.149.0.0 -
            159.149.255.255
netname:    UNIMINET
descr:      Università degli Studi
            di Milano
country:    IT
remarks:    To notify abuse mailto:
            cert@garr.it
remarks:    Multiple-Lans of Milan
            University
```

La conoscenza dell'indirizzo IP del mittente suggerisce l'idea di configurare il proprio MTA di ricezione in modo che possa rifiutarsi di ricevere posta da alcuni server malfamati (*blacklisting*) oppure di accettare connessioni solo da server conosciuti e fidati (*whitelisting*); ma, come vedremo, queste semplici tecniche sono tutt'altro che perfette e possono introdurre ritardi e omissioni di servizio poco graditi agli utenti.

3. COME NASCE LO SPAM

Fino ai primi anni Novanta, la posta elettronica indesiderata consisteva principalmente in innocui scherzi e nei messaggi delle cosiddette "catene di Sant'Antonio". Nell'insieme l'intento di chi li mandava non era criminoso e in pratica non veniva fatto alcun tentativo per falsificare la provenienza dei messaggi, che venivano inviati ai destinatari diretta-

[3] In questo caso viene spesso generato un messaggio di errore, come:
`451 <caio@crema> ... Domain does not resolve.`

mente dal server SMTP del loro mittente. Secondo molti osservatori, la data d'inizio dello spamming commerciale è il 1994, in cui avvenne la diffusione su tutti i gruppi di discussione Usernet del famoso messaggio "*Green-card laywers*" degli avvocati Lawrence Canter e Martha Siegel (Figura 3), che più tardi divennero i primi esperti di Internet marketing. Il messaggio annunciava ai riceventi la fine della lotteria annuale per avere la *Green Card*, il permesso di soggiorno permanente negli Stati Uniti.
Tecnicamente, la novità del "*Green-card laywers*" stava nell'utilizzo di un programma per l'invio sistematico del messaggio a centinaia di gruppi Usenet, e non nella dissimulazione del MTA mittente. Quest'ultimo obiettivo venne raggiunto l'anno successivo da Jeff Slaton, che divenne in breve il primo re dello spam, "*the Spam King*". Nella sua più che decennale attività di *spammer*, Slaton ha affermato di poter raggiungere fino a 8 milioni di persone i cui indirizzi erano entrati in suo possesso grazie alla raccolta su Usenet.
Oggi, lo *spam* è diventato uno dei più grandi problemi dell'Internet moderna e uno spreco in termini di tempo e banda. Secondo un recente rapporto rilasciato dall'agenzia specializzata Sophos (Tabella 1), il 92.3% di tutte le e-mail inviate nei primi tre mesi del 2008 è costituito da *spam*. Stati Uniti e Russia sono in testa alla classifica mentre l'Italia si piazza all'ottavo posto, generando il 3,6% dello spam prodotto nel mondo.

```
Path: gmd.de!urmel.informatik.rwth-aachen.de!newsserver.rrzn.uni-hannover.de!hrz-
ws11.hrz.uni-kassel.de!news.th-darmstadt.de!fauern!zib-
berlin.de!netmbx.de!Germany.EU.net!EU.net!howland.reston.ans.net!europa.eng.gtefs
d.com!hookup!news2.sprintlink.net!news.sprintlink.net!indirect.com!nike
From: n...@indirect.com (Laurence Canter)
Newsgroups: alt.bonehead.paul-hendry,alt.online-service.america-online
Subject: Green Card Lottery- Final One?
Date: 12 Apr 1994 07:40:23 GMT
Organization: Canter & Siegel
Lines: 34
Message-ID: <2odj97$25f@herald.indirect.com>
NNTP-Posting-Host: id1.indirect.com

Green Card Lottery 1994 May Be The Last One!
THE DEADLINE HAS BEEN ANNOUNCED.

The Green Card Lottery is a completely legal program giving away a
certain annual allotment of Green Cards to persons born in certain
countries. The lottery program was scheduled to continue on a
permanent basis.  However, recently, Senator Alan J Simpson
introduced a bill into the U. S. Congress which could end any future
lotteries. THE 1994 LOTTERY IS SCHEDULED TO TAKE PLACE
SOON, BUT IT MAY BE THE VERY LAST ONE.

PERSONS BORN IN MOST COUNTRIES QUALIFY, MANY FOR
FIRST TIME.

The only countries NOT qualifying  are: Mexico; India; P.R. China;
Taiwan, Philippines, North Korea, Canada, United Kingdom (except
Northern Ireland), Jamaica, Domican Republic, El Salvador and
Vietnam.

Lottery registration will take place soon.  55,000 Green Cards will be
given to those who register correctly.  NO JOB IS REQUIRED.

THERE IS A STRICT JUNE DEADLINE. THE TIME TO START IS
NOW!!

For FREE information via Email, send request to
cs...@indirect.com


--
*****************************************************************
Canter & Siegel, Immigration Attorneys
3333 E Camelback Road, Ste 250, Phoenix AZ  85018  USA
cs...@indirect.com   telephone (602)661-3911  Fax (602) 451-7617
```

FIGURA 3
Il messaggio "Green-card laywers" di Canter e Siegel

Paese	Percentuale Spam Prodotta (primo trimestre 2008)
Stati Uniti	15.4%
Russia	7.4%
Turchia	5.9%
Cina	5.5%
Brasile	4.3%
Corea del Sud	4.0%
Polonia	3.8%
Italia	3.6%
Germania	3.4%
Gran Bretagna	3.4%
Spagna	3.3%
Francia	3.1%

TABELLA 1
I Paesi maggiori produttori di Spam (Fonte: SOPHOS)

La raccolta degli indirizzi dei destinatari rappresenta da sempre un problema per gli *spammer* (**Appendice 2** a p. 52), ma gli indirizzi dei server SMTP non sono difficili da trovare: è sufficiente consultare i campi MX presenti nei file di zona DNS. Per combattere i primi *spammer*, i gestori di server SMTP usavano strumenti di *blacklisting* molto semplici, creando in sede di configurazione dei server SMTP una lista (*killfile*) di indirizzi IP dai quali non desideravano ricevere messaggi.
Ben presto però gli *spammer* scoprirono come combattere il *blacklisting* dei loro server, grazie a una funzionalità dei server SMTP chiamata *open relay*. Questa funzione esiste in tutte le implementazioni del protocollo; in questo articolo ci concentreremo su `sendmail`, l'implementazione di SMTP che discende dall'originale DeliverMail di ARPANET[4].
`Sendmail` è ancora oggi il più popolare MTA di Internet, sebbene stia perdendo posizioni. La sua popolarità è probabilmente dovuta al fatto che è il server SMTP standard della maggior parte delle varianti di Unix. Fino alla versione 5, `sendmail` (come molte altre implementazioni di SMTP) inviava messaggi per conto di qualsiasi client lo richiedesse, fungendo appunto da "*open relay*". Invece di spedire lo *spam* direttamente al server SMTP del destinatario, gli *spammer* iniziarono a usare - alternandoli - i server SMTP di altri come intermediari. Ovviamente, l'uso dell'*open relay* non impedisce di per sé il *blacklisting* degli indirizzi IP dei server da cui proviene lo *spam*, ma l'inclusione nelle *blacklist* di server SMTP che fanno *open relay* in buona fede è molto difficile, perché impedisce anche il recapito di messaggi legittimi, che vengono bloccati insieme allo *spam* ritrasmesso.
L'unica vera contromisura sta nel disabilitare la funzionalità di *open relay* su tutti i server SMTP. Questo problema collettivo di configurazione fu oggetto di un grande dibattito nella comunità di Internet, che forse per la prima volta si accorse che un problema tecnico apparentemente banale si poteva tradurre in un incubo organizzativo. Per impedire l'*open relay* basta un semplice script di configurazione per `sendmail` come quello che segue:

```
FR-o /etc/sendmail.cR
Scheck_rcpt
# La posta che va recapitata local-
mente è accettata
R< $+ @ $=w >   $@ OK
R< $+ @ $=R >   $@ OK
# La posta che è generata localmen-
te è accettata
R$*             $: $(dequote "" $&
{client_name} $)
R$=w            $@ OK
R$=R            $@ OK
R$@             $@ OK
# tutto il resto è rifiutato
R$*             $#error $: "550 Re-
laying Denied".
```

Risolvere il problema di quali sono i soggetti organizzativi che hanno titolo ad attivare un server SMTP pubblico e di come garantire il loro comportamento uniforme nella gestione dei server è un'impresa tutt'altro che semplice per le grandi organizzazioni decentrate come le Università. Alla fine degli anni Novanta, comunque, l'azione congiunta dei provider Internet e delle grandi organizzazioni per censire i server SMTP attivi e disabilitare l'op-

4 `Sendmail` è ancora oggi il mail server più usato su Internet. Secondo uno studio del 2005, il 42% circa dei mail server raggiungibili via Internet usavano Sendmail.

zione *open relay* aveva quasi risolto il problema dello *spam*, anche se all'interno delle Università il divieto ai singoli utenti di gestire liberamente il proprio `sendmail` suscitò parecchi malumori[5]. Purtroppo, però, l'evoluzione tecnica della Rete fece presto emergere tre nuove tecniche di recapito che riportarono lo *spam* d'attualità già all'inizio degli anni Duemila.

Recapito Relay multi-hop: il primo fattore è l'aumento di complessità dei servizi di posta elettronica gestiti dai provider, che rese possibile agli *spammer* aggirare il blocco dell'*open relay* attraverso una tecnica detta *relay multihop*. Oggi, infatti, le reti dei provider Internet e delle grandi organizzazioni si affidano a più server SMTP, alcuni usati per l'invio di posta tra utenti dello stesso dominio, ed altri MTA "di confine" usati per inoltrare la posta verso l'esterno. Ovviamente, gli MTA di confine accettano il *relay* da parte dei server interni. Se lo *spammer* ha accesso a uno dei server interni, o se quest'ultimo non è ben configurato, può inviare messaggi di *spam* tramite il MTA di confine, che (pur non facendo *open relay*) accetta di rispedirli verso l'esterno perché gli sembra che provengano da un mittente autorizzato.

Dynamic addressing e recapito No Relay: il secondo fattore è la pratica, oggi prevalente tra i provider Internet, di assegnare ai loro clienti indirizzi IP dinamici, cioè validi solo per la durata di una connessione. Questa prassi ha dato agli *spammer* un altro modo di aggirare il blocco dell'*open relay*: lo *spammer* recapita i messaggi di *spam* direttamente ai server SMTP dei destinatari, usando il suo indirizzo IP dinamico. Periodicamente, oppure ogni volta che l'indirizzo IP dinamico dello *spammer* viene notato e elencato su una *blacklist*, lo *spammer* può semplicemente scollegarsi da Internet, riconnettersi e ricevere un nuovo indirizzo IP dinamico. Il costo di eseguire uno *spamming* di questo tipo è alto anche per lo *spammer* (soprattutto in termini di tempo), ma l'inoltro di *spam* con questa tecnica (detta *no relay*) è molto efficace e combatterlo è estremamente difficile.

Connection Sharing e recapito open proxy: il terzo fattore riguarda la condivisione delle connessioni Internet. Oggi molte organizzazioni usano *proxy* sui loro server connessi a Internet per consentire ad altri computer della loro rete locale (cablata o wireless) di condividere la connessione. Come accadeva i server di posta elettronica che facevano inavvertitamente *open relay*, anche i *proxy software* sono spesso mal configurati e permettono ad *host* "parassiti" di attivare connessioni *proxy* (*open proxy*). Gli *spammer* hanno iniziato a usare i client con open proxy per dissimulare l'origine della posta elettronica. Se un *open proxy* non è disponibile, può essere diffuso in modo virale: già nel gennaio 2003, il noto virus `Sobig.a` installava nei computer vittime un *proxy* concepito specificatamente con l'intenzione di consentire lo *spam*.

Tecniche ibride: per rendere ancora più difficile prendere contromisure, gli *spammer* usano spesso una combinazione delle tecniche appena viste. Per esempio, lo *spammer* usa il server SMTP di un provider Internet poco rigoroso nei controlli o un indirizzo dinamico per raggiungere un server SMTP che fa *open relay*, tramite quest'ultimo, accede al server SMTP di un grosso provider. Il seguente frammento di *header* proviene da un messaggio di *spam* reale:

```
Return-Path: <hymcirrus@coastli-
netrans.com>
Delivered-To: damiani@dti.unimi.it
Received: (qmail 9405 invoked by uid
210); 9 Dec 2008 00:00:03 -0000
Received: from 159.149.10.22 by
pollon (envelope-from
<hymcirrus@coastlinetrans.com>, uid
201) with qmail-scanner-1.25st
 (clamdscan: 0.94.1/8730. spamas-
 sassin: 3.2.1. perlscan: 1.25st.
 Clear:RC:0(159.149.10.22):SA:-
 0(3.8/6.0):.
 Processed in 2.340732 secs); 09 Dec
 2008 00:00:03 -0000
X-Spam-Status: No, hits=3.8 requi-
red=6.0
X-Spam-Level: +++
```

[5] Il timore delle conseguenze dell'attivazione di server SMTP da parte di utenti ignari o inesperti è probabilmente uno dei motivi per cui anche oggi i computer Macintosh vengono consegnati agli acquirenti con `sendmail` disabilitato.

```
Received: from unknown (HELO mail-
server.unimi.it) (159.149.10.22)
  by 0 with SMTP; 9 Dec 2008 00:-
  00:01 -0000
Received: from unimix1.unimi.it
([172.24.4.81])
 by ldap-s2.unimi.net (Sun Java
 System Messaging Server 6.2-8.04
 (built Feb 28
  2007)) with ESMTP id <0KBL004-
  2V1C3BAA0@ldap-s2.unimi.net> for
 damiani@dti.unimi.it (ORCPT er-
 nesto.damiani@unimi.it); Tue,
 09 Dec 2008 01:00:03 +0100 (CET)
Received: from comercigomez.com
(unknown [123.18.210.158]) by uni-
mix1.unimi.it
(Unimi) with ESMT id EFF844A0026 for
 <ernesto.damiani@unimi.it>; Tue,
 09 Dec 2008 01:00:10 +0100 (CET)
```

Qui, come si vede, è stata usata la tecnica ibrida di recapito: sono presenti diversi campi Received: e quindi il messaggio è transitato per diversi MTA. Andando a ritroso troviamo il nostro MTA locale pollon e poi il MTA interno mailserver.unimi.it, di cui pollon si fida e dal quale accetta la posta. Notate che in questo caso il nome del MTA non era presente nel campo From: dell'envelope SMTP ma è stato desunto dal messaggio SMTP HELO con cui mailserver.unimi.it si è annunciato a pollon. Il MTA interno mailserver.unimi.it a sua volta aveva accettato la mail dal server SMTP "di confine" unimix1.unimi.it, l'unico MTA abilitato a parlare con sever esterni.

Proseguendo l'analisi a ritroso la lista dei campi From: troviamo il server SMTP con indirizzo 123.18.210.158, da cui è arrivata la mail. Ecco il potenziale colpevole, dietro cui potrebbe nascondersi lo *spammer*. La figura 4 mostra il risultato della ricerca di questo indirizzo in un database di server SMTP che eseguono *open relay* (http://www.mail-abuse.com).
A questo punto, quindi, la caccia si interrompe: nel parlare con il server open relay, lo *spammer* può inserire i campi Received: che meglio crede nell'intestazione del messaggio di posta, e falsificarli liberamente.

4. IL FILTRAGGIO DEI MESSAGGI

Per neutralizzare le tecniche di *spam* basate sulla tecnica *no relay* (ed alleviare quelle che ricorrono a *open proxy*) si potrebbe in linea di principio adottare il blocco completo degli indirizzi IP dinamici, cioè configurare i server SMTP in modo che non accettino connessioni da altri server che hanno un indirizzo IP dinamico. Si tratta però di un approccio poco pratico perché non esiste un semplice test per stabilire se un indirizzo IP è assegnato dinamicamente o meno[6]. Un'altra tecnica molto interessante è quella delle cosiddette *honeypot*, costituite da finti server SMTP poco scrupolosi e da caselle di posta non corrispondenti a utenti reali, vere e proprie trappole che catturano gli indirizzi IP dei server SMTP usati dagli *spammer*.
In pratica, però, la latenza necessaria per diffondere le segnalazioni delle *honeypot* le rende molto più indicate per attivare contromisure legali che per reazioni in tempo reale

```
The IP address 123.18.210.158 does appear on the following database managed by
Trend Micro's Network Reputation Services.

Database      Entry              Action
DUL           123.18.210.158     Remove

Please see the linked web pages for further information about the database,
contact information, why the address is listed, and how to get it removed, if
applicable.
Please note: These databases are based on IP addresses; they do not use host
or domain names.
```

FIGURA 4
Identificazione del server SMTP open relay

[6] A differenza di quanto alcuni credono, la maggior parte degli indirizzi IP pubblici attribuiti staticamente NON corrisponde a un nome nel DNS. Quindi le *query* inverse al DNS non sono purtroppo un buon test per dedurre la staticità di un indirizzo.

all'invio di *spam*. In generale, il filtraggio dinamico basato sull'IP del server SMTP mittente si è gradualmente rivelato un metodo antispam poco pratico, e all'inizio degli anni Duemila la lotta allo *spam* ha affiancato all'IP *filtering* un'altra direzione, adottando un approccio collaborativo e più orientato al contenuto, sia sui server SMTP, sia sui programmi client usati per spedire e leggere la posta.

4.1. List splitting e personalizzazione dello spam

L'idea iniziale del filtraggio orientato al contenuto fu sfruttare il fatto che la maggior parte degli *spammer* inviava a tutti i destinatari una copia dello stesso messaggio. Facendo quest'ipotesi, il filtraggio collaborativo può funzionare come segue: quando abbastanza utenti di posta segnalano un messaggio sospetto, per esempio mettendolo nella cartella "*Junk Mail*" dei loro client, questi ultimi notificano la cosa al server SMTP e il messaggio incriminato (o meglio una sua rappresentazione compressa, uno *hash MD5*) viene aggiunto a una lista che è poi condivisa tra i server SMTP, con connessioni *peer-to-peer* o attraverso servizi di notifica simili a quelli usati per gli antivirus.
I server SMTP possono poi scartare i messaggi di posta in arrivo il cui *hash* MD5 corrisponde a uno di quelli nella lista dello *spal*. Anche se in un primo tempo il filtraggio collaborativo fu efficace, ci si accorse subito che gli *spammer* potevano aggirarlo usando tecniche di partizionamento e di personalizzazione delle liste dei destinatari (il cosiddetto *list splitting*), in modo da aggiungere ai messaggi di *spam* delle porzioni variabili dipendenti dal destinatario.
In realtà fino ai primi anni Duemila gli strumenti più usati dagli *spammer* non comprendevano funzionalità di "*list splitting*", e molti *spammer* continuarono a inviare a tutti i destinatari gli stessi messaggi fino a quando, nel 2002, non vennero diffusi sul mercato i primi strumenti antispam che usavano classificatori di testo statistici.
L'idea di filtrare i messaggi in arrivo sulla base del loro contenuto non è concettualmente una novità; anzi, fin dagli albori della posta elettronica molti programmi per la lettura della posta sono stati dotati di filtri configurabili in base ai campi dell'intestazione dei messaggi. Queste regole sono in grado di individuare contenuti tipici dei messaggi di *spam*, che non appaiono nei messaggi "normali". Si possono per esempio filtrare i messaggi che non ottengono nel campo `To`: l'indirizzo corretto del destinatario oppure il cui `Subject`: sia vuoto o tutto in maiuscolo, o contenga parole chiave specificate dall'utente. Un altro criterio di filtraggio esamina il campo From:operando analogamente a quanto abbiamo già visto per il campo omonimo dell'*envelope* SMTP. Se il campo `From`: è vuoto, o l'indirizzo del mittente non risponde a certe caratteristiche, il messaggio viene scartato.
Oggi gli amministratori di sistemi Unix hanno a disposizione software ben più evoluti come `procmail` (un programma che processa automaticamente i messaggi quando questi arrivano nella casella locale) per i quali si possono predisporre file di configurazione - e quindi filtri - molto complessi. Uno di questi, *SpamBouncer*, è in grado di generare dei falsi messaggi di errore per far credere allo *spammer* che l'indirizzo a cui si rivolge è inesistente.

4.2. Tecniche automatiche di riconoscimento dello spam

La comunità della ricerca informatica – compreso chi scrive – ha versato in questi anni fiumi d'inchiostro sulle tecniche automatiche di riconoscimento dello *spam*, proponendo diversi algoritmi molto ingegnosi, in grado di classificare messaggi di testo come *spam* in modo rapido ed efficace. Queste tecniche sono in grado di ridurre i falsi positivi (cioè i messaggi che non sono *spam* ma vengono identificati come tali) anche in presenza di *list splitting* e di personalizzazione dinamica del testo dei messaggi di *spam*. Molti *spammer* hanno reagito a queste tecniche evolute di riconoscimento semplicemente spostando la parte informativa dei loro messaggi all'interno di immagini, da inviare poi come allegati MIME (*Multipurpose Internet Mail Extensions*) o agganciare ai messaggi scrivendoli in formato HTML. È noto che i computer sono molto meno bravi degli umani nel riconoscere il contenuto di immagini; anzi, il fatto che la localizzazione di caratteri all'interno di un'immagine è un problema facile per un utente umano ma difficile per un software è oggi sfruttato da molti siti Web per evitare la compilazione au-

tomatica delle *form*[7]. Gli *spammer* usano esattamente la stessa tecnica: generano immagini contenenti il loro testo e sfidano il programma antispam a trovarlo e riconoscerlo per analizzarlo. Questa tecnica è stata alla base dell'epidemia di *spam* grafico diffusasi a partire dal 2006, in cui il testo dello *spam* è convertito in immagini *raster*. Se le immagini usate dagli *spammer* fossero personalizzate per ciascun destinatario, la tecnica grafica sarebbe quasi impossibile da controbattere. Per fortuna molti *spammer* non hanno il tempo e i mezzi per generare le immagini dinamicamente e per applicare fino in fondo il *list splitting*.

5. SPAMMER ALL'ATTACCO

Ben pochi tra gli *spammer* oggi sono esperti di reti IP o di algoritmi evoluti di riconoscimento di immagini: la maggioranza di loro si serve semplicemente di *toolkit software* liberamente disponibili su Internet. Per acquisire una migliore comprensione del funzionamento di questi strumenti per lo *spam*, esamineremo tre strumenti di invio di massa (*bulk mailing*) molto usati dagli *spammer*. Tutti e tre questi strumenti si basano sugli stessi principi base che abbiamo visto in precedenza: il *list splitting* e la personalizzazione dinamica del contenuto dei messaggi di *spam*, ma sono stati sviluppati nel tempo per controbattere l'effetto del software *anti-spam*. Il terzo ha innovato radicalmente la tecnica di recapito, riducendo il tempo di elaborazione e la larghezza di banda che caratterizzano l'invio di *spam* con le classiche tecniche *open relay* e *open proxy*.

5.1. Dark Mailer

Dark Mailer è un software per Windows che è stato lo strumento preferito di Robert Soloway, un noto *spammer* condannato nel luglio 2008 per frode ed evasione fiscale In Dark Mailer la definizione del contenuto del corpo del messaggio è lasciata interamente allo *spammer*, senza alcun controllo di sintassi o funzione di visualizzazione in anteprima. A causa di ciò, i messaggi inviati da Dark Mailer spesso contengono vistosi errori di ortografia. La struttura e le intestazioni dei messaggi vengono trattati separatamente. Dark Mailer richiede che l'utente specifichi una o più "macrointestazioni" che contengono i campi dell'intestazione e la struttura MIME di vari messaggi di *spam*, e poi seleziona casualmente una di queste macrointestazioni per ogni messaggio di spam che genera.
Dark Mailer può trasmettere i messaggi via SMTP, direttamente in *open proxy* o attraverso un server SMTP *open relay*, oppure via HTTP. Rispetto ad altri strumenti, la trasmissione è tutt'altro che rapida, ma si possono inviare messaggi a più destinatari (tramite i comandi SMTP RCPT) e si possono inviare più messaggi per connessione. Sebbene sia molto facile da usare, Dark Mailer è molto lento e richiede uno *spammer* esperto per scrivere il contenuto del messaggio in modo da passare i filtri *anti-spam*. Anzi, gli utenti di Dark Mailer sono spesso diventati facili obiettivi per altri *spammer*[8].

5.2. Send Safe

Send Safe è uno dei più diffusi ed efficaci strumenti di *spamming* oggi in uso. A differenza di Dark Mailer, Send Safe è stato venduto apertamente dal suo autore Ruslan Ibragimov ed è mantenuto ancora attivo (`http://www.send-safe.com/`). È disponibile in due versioni: un'applicazione autonoma per Windows che gestisce campagne di *spam* e un'edizione aziendale che consiste in una console di gestione basata su Windows e in un programma per l'invio posta elettronica che è disponibile per Windows, Linux e Unix FreeBSD. Le due versioni sono simili nelle funzionalità, ma nell'edizione aziendale il motore di consegna di posta elettronica consente di eseguire recapiti in parallelo aumentare la velocità di recapito. Send Safe ha un sistema di gestione della struttura dei messaggi di *spam* ben più evoluto rispetto a Dark Mai-

[7] Basta generare automaticamente un'immagine che contiene in un punto *random* una breve scritta (magari con caratteri ruotati) e chiedere all'interlocutore remoto di riprodurla nella *form* per tagliare fuori chi compila la *form* tramite uno script.

[8] Gli *spammer* esperti spesso infettano il software Dark Mailer con vari malware prima di passarlo ad altri *spammer* neofiti.

ler. Mentre la configurazione di Dark Mailer supporta un solo *template* per messaggi di *spam*, la configurazione di Send Safe è organizzata in "campagne" e "messaggi".
Una campagna Send Safe consiste in uno o più messaggi e un insieme di *mailing list*. Un messaggio è costituito da un corpo del messaggio e da una serie di argomenti per il campo `Subject:`, indirizzi per il campo From: e allegati. Una campagna invia periodicamente i suoi messaggi a tutti indirizzi contenuti nei file delle mailing list.
Come Dark Mailer, Send Safe consente la trasmissione diretta di messaggi basata su *open proxy* e *open relay*, ma applica alcune tecniche evolute. Per eludere le *black list*, Send Safe può cambiare continuamente l'indirizzo IP che usa per collegarsi ai server di posta elettronica o ai *proxy*. Send Safe dispone anche di un *proxy* interno che e' stato progettato per eludere l'individuazione tramite *honeypot*. Invece di connettersi direttamente alla lista di *proxy* specificata dallo *spammer*, si collega ad essi attraverso una serie di *proxy* intermedi considerati sicuri. Se c'è un *honeypot* nella lista di *proxy* dello *spammer*, l'indirizzo IP del sistema su cui gira Send Safe non sarà compromesso.
Un'altra tecnica interessante introdotta da SendSafe è il *proxy locking*. Partendo dall'indirizzo IP di un open *proxy*, Send Safe usa una query DNS inversa per cercare nel record `MX` (`Mail Exchanger`) il server SMTP usato dal *proxy*. Invece di tentare di consegnare i messaggi attraverso il *proxy*, SendSafe si rivolge direttamente al server SMTP. Questo trucco può portare i server SMTP di produzione dei provider a comparire gli uni nelle *black list* degli altri. La contromisura più evidente è attivare il filtraggio orientato al contenuto dello *spam* di cui abbiamo parlato prima anche in uscita (e non solo in ingresso) dai server SMTP interni, ma questo ha costi non indifferenti e introduce sensibili latenze nel recapito della posta.
Send Safe comprende un sistema avanzato per creare *template* di messaggi di *spam*. Si possono generare messaggi che sembrano inviati da client di posta elettronica diversi, come Microsoft Outlook Express e Mozilla Thunderbird. Quando Send Safe invia lo *spam*, alterna i *template* così che ogni messaggio successivo che viene inviato sembra essere stato spedito usando un client diverso.
Send Safe comprende anche diverse contromisure per ingannare i filtri antispam orientati al contenuto. Per esempio, può aggiungere contenuto casuale nei campi `Subject:` e `From:`, oppure codificare la parte testuale (tipo `MIME text/html`) del messaggio usando il codice `base64` invece del `quoted-printable` standard, o ancora aggiungere in modo casuale dei tag HTML al testo del messaggio per confondere i parser HTML di alcuni filtri *anti-spam*. Ben più importante è la capacità di Send Safe di applicare algoritmi di *morphing* alle immagini per deformarle, in modo che non siano facilmente riconoscibili da eventuali algoritmi di classificazione delle bitmap. La generazione delle immagini è però lasciata allo *spammer*, e quindi Send Safe non è molto adatto per le campagne di *spam* grafico che fanno forte ricorso al *list splitting* e personalizzano i messaggi.

5.3. Reactor Mailer

Reactor Mailer, venduto dalla società ucraina Elphisoft, è di gran lunga il sistema di *spamming* più interessante sviluppato fino ad oggi. Mentre Dark Mailer e Send Safe generano i messaggi di *spam* localmente e poi li trasmettono attraverso una lista di *open proxy* e server SMTP che accettano *open relay*, Reactor Mailer usa un modello computazionale distribuito simile a quello dei virus. Il programma si compone di un server e di un client distribuito in forma virale, che gli antivirus Symantec conoscono come `Trojan.Srizbi`. I personal computer che vengono infettati dal client Reactor Mailer scaricano periodicamente *template* di messaggi e liste di indirizzi di posta elettronica, generano e trasmettono indipendentemente i loro messaggi e poi rimandano i report dei risultati al server. Questa tecnica riduce molto i costi di tempo di elaborazione e di larghezza di banda che rendono oneroso l'invio di *spam* tramite Dark Mailer e Send Safe.
Reactor Mailer usa un sistema di *template* simile al sistema di intestazioni di Dark Mailer; il *template* più usato crea messaggi quasi indistinguibili da quelli generati da Outlook Express 6.
Mentre Send Safe richiede che l'utente crei le

proprie immagini, Reactor Mailer comprende la traduzione del testo dello *spam* a immagine. Questo sistema può creare immagini basate su testo formattato HTML e può offuscare le immagini attraverso l'aggiunta di rumore *random* e rototraslazioni dei caratteri.

6. Un esempio

Vediamo ora una versione semplificata di un *template* di Reactor Mailer:

```
From: {rndline 008_wname.txt}-
{rndabc 1}@{rndline
003_domains.txt}
Subject: {rndline 001_subject.txt}
{rndline 005_hi.txt}

{rndline 001_msg.txt}
http://{rndline 006_sub.txt}.-
{rndline 000_067.txt}
{rndline 004_fin.txt}
{rndline 002_afo.txt}, {rndline
002_afo.txt}
```

Le intestazioni dei messaggi di *spam* generate usando questo *template* contengono un campo `From`: generato a caso, un nome di battesimo e l'iniziale di un cognome casuali come username e un `Subject`: anch'esso selezionato a caso da una lista. Il corpo del messaggio inizia con un saluto scelto a caso da una lista e poi continua con una frase scelta a caso da una terza lista. Le frasi sono seguite da un URL *random* e poi il messaggio si conclude un saluto scelto a caso. Questo *template* può produrre un numero elevatissimo di messaggi diversi, rendendo difficile il lavoro dei filtri antispam orientati al contenuto. Ecco un esempio di *spam* generato dal *template*:

```
From: LombrosoC@pollon.it
Subject: Chi dorme non piglia pesci
Come butta oggi?
Le brave ragazze vanno in Paradi-
so, le cattive
dappertutto.
http://vieniacasa.org
Grazie per l'attenzione, gente!
La svelta volpe balza sul cane pi-
gro, non aspettate tempi
migliori.
```

7. LE CONTROMISURE

Vediamo ora le contromisure che possono essere prese contro lo *spam* usando gli strumenti di difesa basati sulle tecniche che abbiamo spiegato all'inizio dell'articolo. La soluzione di riferimento è SpamAssassin, un software che identifica automaticamente lo *spam*. Pur essendo pensato per sistemi Unix, grazie al fatto di essere open source SpamAssassin è stato proposto anche come add-in per alcuni mail server commerciali. Per identificare lo *spam* SpamAssassin esegue una serie di verifiche sull'intestazione e un'analisi del testo del messaggio. Inoltre, usa alcune blacklist di MTA inaffidabili reperibili in Rete. Dopo essere stato identificato, lo *spam* viene contrassegnato con un punteggio che si aggiunge all'intestazione del messaggio, in modo che quest'ultimo possa poi essere filtrato dal client di posta dell'utente.
Ecco un esempio dell'aggiunta generata da SpamAssassin:

```
spamassassin: 3.2.1. perlscan:
1.25st.
 Clear:RC:0(159.149.10.22):SA:-
 0(3.8/6.0):.
 Processed in 2.340732 secs); 09 Dec
 2008 00:00:03 -0000
X-Spam-Status: No, hits=3.8 requi-
red=6.0
X-Spam-Level: +++
```

Per gli esempi di *spam* SpamAssassin si basa su Vipul's Razor, una rete distribuita e collaborativa di identificazione dello *spam* che opera da un paio d'anni, grazie alla quale è stato costruito un catalogo costantemente aggiornato dello *spam* in circolazione. Lo strumento Spam Arrest, invece adotta un approccio basato su *whitelist*, una lista di "amici" autorizzati a scriverci. Se qualcuno che non è nella lista scrive a una mailbox protetta da Spam Arrest, riceverà immediatamente un messaggio che lo invita a visitare un sito, da cui può iscriversi alla lista di amici. Per poterlo fare, dovrà trascrivere in un campo testo il contenuto di un'immagine che riporta caratteri testuali in posizione *random*, dimostrando così di essere una persona e non uno script utilizzato da uno *spammer*. Veniamo ora a due tecniche "storiche" che per i motivi pratici esposti fin qui non hanno risolto il

problema dello *spam*, ma risultano comunque particolarmente interessanti: il *reverse spam filtering* e i filtri bayesiani.

7.1. Reverse Spam Filtering

La strategia del *Reverse Spam Filtering* è diametralmente opposta a quella dei filtri orientati al contenuto. Questa tecnica infatti si propone di selezionare ciò che NON è *spam* e mandare tutto il resto in una cartella speciale, che viene controllata solo periodicamente. Anzitutto il sistema controlla se il messaggio in entrata appartiene a qualche invio di massa sollecitato (*mailing list o newsletter*). In questo caso viene messo in un'apposita cartella. Altrimenti, viene controllata la provenienza: se il messaggio proviene da indirizzi approvati (cioè definiti in una lista di "amici" come quella di SpamArrest) viene posto in un'apposita cartella altrimenti il messaggio viene analizzato e quindi marchiato come *spam* con una certa probabilità, e inserito in una speciale cartella per i messaggi sospetti, il cui contenuto può essere ordinato in base alla probabilità e controllato manualmente per cercare falsi positivi. Il *Reverse Spam Filtering* necessita di un software per filtrare i messaggi, uno per analizzare e assegnare un punteggio di probabilità ai messaggi sospettati di essere *spam*, un buon client di posta che permetta di gestire più mailbox e di ordinare il contenuto delle mailbox in base a criteri personalizzati, un sistema per mantenere facilmente o automaticamente una lista di indirizzi "amici" aggiornata. In genere si usa `procmail` per filtrare i messaggi in arrivo e SpamAssassin per marchiare i messaggi con un punteggio di *spam*.

7.2. Filtri bayesiani

La soluzione bayesiana è stata proposta inizialmente da Paul Graham ed è basata sullo studio statistico del contenuto dei messaggi. Un filtro bayesiano decide se un messaggio è *spam* o no in base alle parole contenute nei messaggi ricevuti da uno specifico utente.
Prima di illustrare l'algoritmo usiamo un semplice esempio per ricordare il teorema di Bayes: abbiamo un'osservazione O (un messaggio contiene la parola "sex") e un'ipotesi H (un messaggio è *spam*). P(O|H), cioè la probabilità che O accada dato H, ovvero la probabilità che un messaggio di *spam* contenga la parola "sex", è facile da stimare (ad esempio esaminando la cartella "Junk Mail" in cui l'utente destinatario mette lo *spam* e contando quanti dei messaggi che vi si trovano già contengono "sex"). Per il futuro, ci interessa però sapere P(H|O), cioè la probabilità che H accada, dato O, e cioè che un messaggio indirizzato a quell'utente e che contiene la parola "sex" sia effettivamente *spam*. Secondo il teorema di Bayes tale probabilità è:

$$P(H|O) = P(O|H) * P(H)/P(O)$$

Dove sia P(H) (la probabilità che un messaggio sia *spam*) sia P(O) (la probabilità che un messaggio contenga la parola "sex") possono essere agevolmente stimate esaminando le caselle di posta dell'utente. P(H) si stima esaminando comparativamente la cartella "Junk Mail" dove l'utente mette lo *spam* e la casella di posta generale dell'utente e contando quanti sono i messaggi di *spam* rispetto al totale dei messaggi. P(O) si stima contando quanti messaggi contengono "sex" sul totale dei messaggi (*spam* o no) ricevuti dall'utente. Va notato che queste probabilità devono essere calcolate per ogni utente perché, se i messaggi di spam possono essere simili per tutti (e a volte sono esattamente gli stessi), quelli personali sono invece molti diversi, e il filtro bayesiano ne tiene automaticamente conto. Le esperienze di Graham, e degli altri ricercatori che hanno lavorato nel settore, ci dicono che il suo filtro è esatto al punto di mancare solo 5 messaggi di spam ogni 1000, senza alcun falso positivo. Rispetto ai filtri visti in precedenza, che funzionano in base alle proprietà individuali di un singolo messaggio, l'approccio statistico su insiemi di messaggi è migliore, perché tiene conto delle specificità dei singoli utenti, esattamente come fa lo *spammer* applicando il *list splitting*. Purtroppo però questa tecnica è praticamente impotente contro lo *spam* grafico.

8. PROTEZIONE CRITTOGRAFICA DEGLI INDIRIZZI

Una prospettiva integralmente nuova è invece quella di togliere agli *spammer* la loro "benzina", cioè gli indirizzi di posta elettronica, attraverso nuovi schemi di generazione dinamica degli indirizzi di posta.
Gli *spammer* usano programmi appositi (det-

```
<script type="text/javascript" language="javascript">
<!--
// Email obfuscator script 2.1 by Tim Williams, University of Arizona
// Random encryption key feature by Andrew Moulden, Site Engineering Ltd
// This code is freeware provided these four comment lines remain intact
// A wizard to generate this code is at http://www.jottings.com/obfuscator/
{ coded = "o8Sm8cm@oLm.McmSm.mL"
  key = "bdyPfKCQwJMAFtVNSD4Oso6pz5X0kGIahLnxUvi3W9ejrZ7EH2l8uYcRTBmgq1"
  shift=coded.length
  link=""
  for (i=0; i<coded.length; i++) {
    if (key.indexOf(coded.charAt(i))==-1) {
      ltr = coded.charAt(i)
      link += (ltr)
    }
    else {
      ltr = (key.indexOf(coded.charAt(i))-shift+key.length) % key.length
      link += (key.charAt(ltr))
    }
  }
document.write("<a href='mailto:"+link+"'>inviate mail al docente</a>")
}
//-->
</script><noscript>Sorry, you need Javascript on to email me.</noscript>
```

FIGURA 5
Uno script che genera l'indirizzo damiani@dti.unimi.it

ti *harvester* o *spambot*) che scaricano le pagine Web alla ricerca di indirizzi di posta a cui mandare *spam* (vedi Appendice 2).
Alcuni siti usano già oggi delle semplici precauzioni per evitarlo, pubblicando indirizzi "antispam" come `ernesto.damiani AT unimi DOT it`. Questo metodo però richiede che sia il visitatore umano a modificare l'indirizzo di posta per renderlo usabile; inoltre è facilmente aggirato dagli spambot più recenti.
Altri siti cercano di difendersi dagli harvester con le loro stesse armi, ossia pubblicando immagini che mostrano gli indirizzi di mail in luogo degli indirizzi un formato testo; ma anche questa precauzione può non essere gradita ai visitatori umani, che devono ridigitare l'indirizzo da capo per poterlo usare.
Infine, altri siti usano la codifica carattere per carattere HTML, per esempio usando @ per il carattere chiocciola (@), o per tutti i caratteri dell'indirizzo. Ecco la codifica dell'indirizzo `someone@example.com`:

```
&#115;&#111;&#109;&#101;&#111;
&#110;&#101;&#64;&#101;&#120;
&#97;&#109;&#112;&#108;&#101;
&#46;&#99;&#111;&#109;
```

Questo tipo di codifica è anch'esso facile preda degli harvester, perché qui ogni carattere corrisponde esattamente a un codice secondo una tabella ben nota. Le tecniche crittografiche invece si basano su una codifica crittata dell'indirizzo di mail. Questa codifica viene decrittata da uno script JavaScript solo al momento dell'utilizzo dell'indirizzo di posta elettronica e quindi quest'ultimo non compare da nessuna parte nella pagina.
Un esempio di uno script di questo tipo per l'indirizzo `damiani@dti.unimi.it` è riportato nella figura 5 – come si vede, si tratta di un osso piuttosto duro per qualunque analizzatore di codice.
Attualmente la tecnica crittografica per la generazione dinamica degli indirizzi di mail viene complementata dalla messa a punto di strumenti innovativi per la generazione e la gestione di "*short-lived alias*", cioè indirizzi di posta monouso. L'idea è di dare ai diversi interlocutori che possono contattarci indirizzi di mail diversi, alcuni dei quali consentono di contattare il destinatario una volta sola.
Nel futuro avremo quindi due tipi di indirizzi: quello stabile (*master*) e quelli temporanei (*alias*). Quando contatteremo qualcuno per la prima volta al suo indirizzo master, gli manderemo un nostro alias. L'interlocutore invierà la risposta al nostro alias, corredando

il messaggio di risposta di un suo alias, e da quel momento potremo proseguire a comunicare usando gli alias monouso allegati a ogni messaggio.
Per rimanere compatibili con il normale recapito SMTP, gli alias avranno sempre l'indirizzo master come suffisso, come segue:

`2BEF9.damiani@dti.unimi.it`

Gli alias potranno così essere risolti sugli MTA di recapito. Gli alias monouso sono la nostra principale speranza di liberarci definitivamente degli *spammer*: valgono per una sola consegna e rendono molto più costoso e difficile il lavoro dello *spammer*. Ovviamente è possibile pensare a varie categorie di alias, magari accettabili più volte o da un gruppo di mittenti predefinito.

9. CONCLUSIONI

È abbastanza chiaro che gli attuali algoritmi di individuazione e filtraggio dello *spam* basati sul contenuto dei messaggi hanno efficacia limitata se i messaggi sono grafici e/o personalizzati rispetto ciascun destinatario. Oggi gli *spammer* hanno a disposizione gli strumenti (se non la conoscenza) per realizzare *template* di messaggi che possono creare un numero elevatissimo di messaggi univoci. Il numero delle permutazioni che possono essere prodotte da questi strumenti è sufficiente per sopraffare i sistemi tradizionali *antispam*, per quanto ingegnosi siano gli algoritmi di classificazione che utilizzano. A volte l'aggiunta ai sistemi antispam di precauzioni semplici, come proibire del tutto il recapito di immagini bitmap come allegati, può migliorarne notevolmente l'efficacia, ma non c'è alcun dubbio che – in attesa di tecniche crittografiche veramente efficaci per la generazione e la risoluzione di indirizzi monouso – il vantaggio resta, almeno per ora, dalla parte degli *spammer*. Gli strumenti per creare *spam* guidati da *template* hanno raggiunto una certa maturità, e la tecnologia *antispam* deve quindi migliorare. Per quanto riguarda il filtraggio del contenuto, allo studio ci sono nuove tecniche statistiche e di apprendimento computazionale che utilizzano la regolarità tipiche dei messaggi generati a partire da *template* invece di concentrarsi, come quelli attuali, sulle regolarità tipiche dei messaggi scritti a mano. La battaglia tra *spammer* e tecniche antispam non è comunque destinata a terminare tanto presto.

APPENDICE 1 - *ASPETTI NORMATIVI E LEGALI*

Il primo Paese a prendere contromisure normative contro lo *spam* sono stati gli USA, che sulla base di una legge federale già in vigore contro l'abuso dei fax, diedero vita alla CAUCE (*Cohalition Against Unsoliticed Commercial Email*), per porre rimedio al vuoto legislativo in materia di e-mail non richieste. Questo compito richiese molto tempo, anche per la continua controffensiva degli *spammer* che premevano per legalizzare l'*opt-out* (ossia la possibilità di negare l'invio di e-mail non richieste solo dopo averle ricevute).

Nel 2003 finalmente il Congresso americano varò la nuova legge federale "*CAN-SPAM Act of 2003*". Questa legge si fondava sul principio dell'opt-out e attribuiva il titolo di agire contro gli *spammer* ai soli Internet provider, e non agli utenti finali dei servizi di posta.

In Europa furono fatti vari tentativi per giungere ad una legislazione comune. Il risultato fu la Direttiva 2002/58/CE del Parlamento Europeo e del Consiglio del 12 luglio 2002, che costituì l'obbligo per gli Stati aderenti alla Comunità Europea di emanare provvedimenti legislativi sul principio dell'*opt-in* e quindi del preventivo consenso del destinatario.

In Italia la principale fonte normativa sull'argomento è la legge 675/96 sulla protezione dei dati personali. L'indirizzo di posta elettronica è considerato come un dato personale, anche se non contiene il nome del titolare. La legge sulla privacy non vieta direttamente l'invio di posta commerciale, ma limita l'uso dell'indirizzo di posta elettronica in determinati casi. Un principio importante è che gli indirizzi e-mail reperibili su internet non sono pubblici e non possono essere usati per fini com-

segue

merciali. Non basta quindi, per poter considerare pubblico un indirizzo di e-mail, il fatto che tale indirizzo sia conoscibile, in determinate circostanze, da una pluralità di persone come può succedere per un indirizzo pubblicato su Internet. Inoltre non possono essere considerati pubblici neanche gli indirizzi di e-mail che vengono pubblicati su forum o newsgroup. Gli indirizzi e-mail in rete possono essere utilizzati solo per le finalità che hanno portato alla loro pubblicazione. Questo principio rende pertanto non conformi alla legge né la raccolta automatica di indirizzi di e-mail presenti su internet né la loro creazione artificiosa, attività che si possono realizzare oggi con appositi software. Inoltre, la legge obbliga le persone fisiche o giuridiche a cui sono stati consegnati i dati, a fornire una descrizione chiara e precisa di quale uso ne verrà fatto: lettura, memorizzazione, trasferimento a terze parti, comunicazioni di servizio o comunicazioni commerciali; inoltre nel momento in cui si forniscono i dati, o in qualunque momento successivo, i titolari dei dati hanno il diritto di sapere entro 5 giorni dalla richiesta in quali termini verranno utilizzati o anche di limitarne o proibirne completamente l'uso.

Questo elemento è molto importante perché neutralizza la difesa degli *spammer* che si basa sulla classificazione degli indirizzi di posta elettronica reperiti sul web come pubblici. È possibile quindi perseguire contro gli *spammer* già grazie alla legge 675/96 anche se in realtà il procedimento si rivela lungo e costoso e soprattutto riguarda solo gli *spammer* italiani.

Sono state poi varate anche legislazioni più specifiche in materia. Per primo il decreto legislativo 171 del 1998, il quale sancisce che il costo pubblicitario deve essere sostenuto interamente da chi fa la pubblicità e non da chi la subisce. Da segnalare anche il decreto legislativo n.185 del 22 maggio 1999 che, quando ancora la Comunità Europea non si era espressa in materia, schierò l'Italia sul fronte *opt-in*. Dopo una serie di interventi mirati alla sospensione di attività illecite o alla denuncia all'autorità giudiziaria di talune aziende o persone fisiche il Garante della privacy è sceso in campo in maniera chiara e dettagliata per disciplinare l'argomento. Il decreto legislativo 30 giugno 2003 n. 196, denominato "Codice in materia di protezione dei dati personali", entrato in vigore dall'1 gennaio 2004, infatti, recepì nell'ordinamento italiano la direttiva europea 2002/58/CE e precisò vari aspetti legali riguardanti l'invio in Internet di e-mail promozionali o pubblicitarie.

Appendice 2 - La raccolta di indirizzi

Gli spammer usano diverse tecniche per recuperare gli indirizzi di posta a cui inviare i loro messaggi indesiderati.

Le principali sono elencate di seguito:

- *Dictionary attack*: questa tecnica molto diffusa si basa semplicemente sull'indovinare gli indirizzi. Più precisamente lo *spammer* cerca di comporre e generare indirizzi che potrebbero effettivamente esistere. Per la parte destra della chiocciola (@) usa nomi di dominio validi e per la parte sinistra genera stringhe in base a qualche logica, per lo più nomi di persone. Per questo motivo l'indirizzo nome.cognome@dominio.it è uno dei più soggetti a questo tipo di attacco.
- *Address list*: un secondo sistema consiste nell'acquisire liste di indirizzi da soggetti che li raccolgono per poi rivenderli. Le liste di indirizzi selezionate, per esempio, sull'attività professionale del destinatario vengono vendute a prezzi elevati, che possono arrivare a diversi dollari per indirizzo nel caso di medici e commercialisti.
- *Spambot*: come abbiamo visto nell'articolo, uno *spambot* o *harvester* è un particolare tipo di *web-crawler* in grado di raccogliere gli indirizzi e-mail dai siti web, dai newsgroup, dai post dei gruppi di discussione e dalle conversazioni delle *chat-room*. Gli basano sullo stesso principio del funzionamento degli spider dei motori di ricerca, ma a differenza di questi ultimi estraggono dalle pagine web tutti gli indirizzi presenti.

BIBLIOGRAFIA

[1] Nancy McGough: Reverse spam filtering - Winning Without Fighting, 4 settembre 2002. In: Infinite Ink, http://www.ii.com/internet/messaging/-spam/ (consultato il 2 dicembre 2008).

[2] Paul Graham, A plan for spam, http://www.paulgraham.com/spam.html

[3] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi Pierangela Samarati, Andrea Tironi, Luca Zaniboni: *Spam attacks: P2P to the rescue*. Proceedings of the 13-th international conference on World Wide Web (WWW 2004), 2004.

Siti interessanti

SpamAssassin, http://eu.spamassassin.org/ .

Vipul's Razor, http://razor.sourceforge.net/ .

Cloudmark, http://www.cloudmark.com/ .

Despammed, http://www.despammed.com/ .

Spamex, http://www.spamex.com/ .

Spam Arrest, http://www.spamarrest.com/

ERNESTO DAMIANI si occupa di sicurezza nei Web services, processing di informazioni semi o non strutturate, semanticsaware content engineering per il multimedia, modelli e piattaforme per lo sviluppo di codice open source, infrastrutture e protocolli di rete avanzati, design e sviluppo di ambienti di rete sicuri ad alte prestazioni. È membro di numerosi editorial boards e ha pubblicato numerosi libri e circa 200 articoli scientifici oltre a brevetti internazionali. Collabora all'organizzazione di molti congressi, conferenze e workshop.
E-mail: ernesto.damiani@unimi.it