



- 1 ¿Qué es el SPAM?
- 2 Consideraciones para luchar contra el SPAM
- 3 Como luchar activamente contra el SPAM
 - 3.1 Análisis del contenido
 - 3.2 listas negras
 - 3.3 Filtro de firma
 - 3.4 SpamPots
 - 3.5 Listas grises
- 4 Puntuando los diferentes métodos de detección
- 5 Referencias

¿QUÉ ES EL SPAM?

Las ventajas del correo electrónico han hecho que algunas empresas que quieren hacer publicidad lo usen indiscriminadamente. Como el coste de enviar correos es casi nulo, los mensajes llegan muy rápido, y lo hacen independientemente de dónde se encuentre físicamente su destinatario, se puede hacer llegar el mensaje publicitario a 200 millones de personas con un coste muy bajo. Estos mensajes se llaman SPAM, o más formalmente UBE (Unsolicited Bulk Email).

Normalmente ignoramos estos mensajes, pero igualmente causan las molestias de descargarlos, detectarlos y eliminarlos. Se estima que alrededor del 20% del correo que circula por Internet es SPAM, y que se pierde mucho tiempo si lo tratan las personas, por tanto un filtro que se deshaga del máximo de SPAM posible es necesario.

CONSIDERACIONES PARA LUCHAR CONTRA EL SPAM

El SPAM es difícil de detectar [3], principalmente porque es adaptativo [4]. Esto significa que los administradores estudian el SPAM para encontrar la manera de evitarlo, pero las personas que envían el SPAM también estudian los métodos de los administradores para poder pasar por encima de ellos, y hacer llegar sus mensajes. Es la historia de la jirafa y la copa del árbol: el árbol crece para no tener la copa al alcance de la jirafa, y la jirafa va evolucionando, cada generación con el cuello más largo, para llegar a comerse las hojas. Cada vez que se idea un nuevo sistema anti-spam, hay un tiempo de tranquilidad donde el spam se reduce, hasta que los spammers encuentran la manera de saltárselo, y así sucesivamente.

El problema del correo no deseado se tiene que afrontar con mucho cuidado [5], porque un filtro demasiado permisivo no parará el spam, pero un filtro demasiado restrictivo hará que existan los 'falsos positivos' [6], correos legítimos que no han llegado a su destino porque se han confundido con spam. Esto es muy importante, no se puede perder ni un correo legítimo: es preferible que haya 1.000 correos no deseados que un solo falso positivo. Por tanto, tenemos que encontrar un punto de compromiso a la agresividad de los filtros tal que garantice que los mensajes legítimos lleguen siempre, y que llegue el mínimo spam posible.

Para evitar al máximo los daños que puedan causar los 'falsos positivos', podemos responder de forma automática a los mensajes considerados SPAM, diciendo que el mensaje se ha rechazado y explicando los motivos. Así cuando alguien enviara un mensaje legítimo, y este fuera rechazado, el remitente lo sabría y podría volver a intentarlo cambiando un poco el mensaje, o intentar enviarlo de otra manera. Parece importante que el remitente sepa que el mensaje no se ha podido distribuir, para que se puedan tomar las medidas pertinentes, pero esto comporta muchos problemas.

En primer lugar, si consideramos un 5% de 'falsos positivos' (aunque comprobaremos que serán muchos menos), vemos que nuestro servidor tiene que enviar correos inútiles el 95% de las veces. Estos mails suponen un coste, y no aportan nada. No solo no aportan nada, sino que pueden dar al spammer una forma de averiguar que correos atraviesan el filtro y cuáles no, y repetir los que no han pasado variándolos hasta conseguir que pasen. Esto supondría mucha más carga para nuestro servidor.

Por otro lado, el servidor de correo saliente (MSA) nunca comprueba la dirección del remitente. Por tanto, cualquier persona puede enviar en cualquier momento un correo electrónico poniendo como remitente una dirección de correo que no es la suya [7]. Los spammers usan listas de direcciones de correo 'víctimas' donde hay un porcentaje alto de direcciones incorrectas/inexistentes, y lo saben. Saben que cuando envían correos a direcciones incorrectas les devolverán mensajes de error, y eso no les interesa porque les gasta ancho de banda, y lo tienen que pagar sin amortizarlo. Por tanto los spammers usan direcciones falsificadas (forged) como remitentes de sus mensajes, y de esta manera no tienen que revelar su identidad y se ahorran el tráfico que no les interesa [8] (los mails que informan de que la dirección no es válida llegarán a los falsos remitentes). Es tan frecuente esta manera de actuar que esta bautizada, el nombre es "JoeJob" [9] [10]. También se usa como ataque DoS distribuido [11].

Con estos datos podemos resumir que si respondemos los mensajes, damos a los spammers una posibilidad de retroalimentación que, si la aprovechan, puede servir para mejorar sus ataques, gastamos ancho de banda inútilmente el 95% de las veces, y además es muy probable que respondamos a gente que no tiene nada que ver con el mensaje que provoca la respuesta. Por tanto no parece muy acertado. La mejor solución es hacer como si el mensaje hubiera pasado correctamente, y hacer todo lo posible para evitar los 'falsos positivos'.

CÓMO LUCHAR ACTIVAMENTE CONTRA EL SPAM

Las maneras de detectar el spam que se han usado hasta ahora son las siguientes [12]:

3.1.- Análisis de contenido

► Análisis del contenido [13]: Este método analiza el cuerpo del mensaje buscando los rasgos comunes del correo no deseado, como por ejemplo palabras que se repiten mucho (nombres de drogas como valium, xanax) o cabeceras de mail inválidas, entre otros. Funciona dentro del servidor de correo electrónico (MTA), una vez se ha establecido la conexión y se ha transferido el mensaje. Al principio este método se configuraba manualmente, y por eso era muy poco fiable, ya que los spammers solo tienen que evitar repetir patrones. Por ejemplo cambiar letras (vaalium vallum, va l i um ...).

Más adelante aparecieron los filtros bayesianos [14], que intentan aprender de los correos que se saltan el filtro. Cuando un correo no deseado llega al buzón, la persona que lo lee, una vez detecta que es spam, se lo devuelve al servidor diciéndole que eso es spam. El servidor lo analiza y mediante algoritmos que implementan el Teorema de Bayes [15] se reestructura con la nueva información para intentar que los mensajes que se le parezcan no vuelvan a pasar. Así los filtros se van adaptando a las nuevas formas de actuar de los spammers

Pero esto se complicó cuando los spammers atacaron directamente estos filtros enviando correos que, después del mensaje de propaganda, contenían palabras de uso frecuente, colocadas aleatoriamente [16]. De esta manera conseguían que los filtros bayesianos, al ser retroalimentados con estos correos, confundiesen algunas palabras de uso frecuente con spam, consiguiendo que correos legítimos, que con mucha probabilidad contienen estas palabras, se convirtieran en 'falsos positivos'. Y así se perdió gran parte de la potencia de este sistema, cuando los 'falsos positivos' fueron demasiado numerosos. Ya no se podía usar como filtro absoluto, aunque aun hay gente que afirma lo contrario [17].

3.2.- Listas Negras

► Listas negras [18]: Hasta principios del siglo XXI el spam no se consideró un problema grave, y había mucha indiferencia entre los administradores de los servidores de correo. Como controlar quién enviaba correos a través de su servidor era un problema a resolver, lo obviaban, y abundaban los servidores de correo que permitían el envío de correos sin ninguna restricción a todos los usuarios que quisieran. Estos servidores se bautizaron como OPEN RELAYS [19]. En estos servidores los spammers encontraron una forma de ahorrarse mucho tráfico: si usan su propio servidor de correo, y envían un mensaje a 2.000 destinatarios, el tráfico resultante de enviar el mensaje es el tamaño del mismo multiplicado por el número de destinatarios. En cambio, si envían un solo mensaje con 2.000 destinatarios en la cabecera, a través de un OPEN RELAY, el coste para el spammer es sólo el tamaño del mensaje, que se transfiere una sola vez al OPEN RELAY, y es este quien se encarga de distribuirlo, teniendo el coste real del tráfico de los 2.000 mensajes.

Como los administradores no se preocupaban de solucionar este problema, y cada vez era más grave, se buscó una forma de 'forzarlos', y se idearon las 'listas negras'.

El método de listas negras consiste en mantener un listado de servidores de correo de los que se tiene constancia que envían spam, marcándolos como 'indeseables'. Nuestro servidor de correo (MTA), al recibir una conexión, comprueba si el servidor remitente está en la lista negra o no. Si está, cierra la conexión y no acepta el correo. Por tanto, funciona antes de la transmisión del mensaje, ahorrándose el tráfico de los correos no deseados.

Con este método, si se extendía lo suficiente, se 'marginaba' a los servidores OPEN RELAY,

que ya no tenían sólo el problema del SPAM, sino que se les añadía el problema que el resto de servidores de correo no aceptaban sus mensajes legítimos. O sea, que si querían poder enviar mensajes a todas partes, como se supone que tiene que funcionar, tenían que hacer que les sacara de las listas negras. Y para eso tenían que cerrar el acceso indiscriminado del servidor, y notificarlo a la lista que los había listado, para que lo comprobasen, y los quitaran si procediera.

Como la lista de servidores 'no deseables' era tan dinámica, aparecieron empresas que se dedican a detectar los OPEN RELAY y ofrecer a los servidores de correo una forma de comprobar si una dirección está en la lista o no [20]. Esta comprobación se tenía que integrar dentro de todos los MTA, y tenía que ser muy rápida y usar muy poco tráfico. Se hacía a través de resolución DNS inversa o a través de una consulta directa.

Una vez se vio que este sistema funcionaba, empezaron a aparecer alternativas colaborativas (rbl, dnsrbl, spamcop, etc [21]), que no dependían de una empresa sino que los propios administradores de los servidores de correo se encargaban de mantener: cuando recibían un correo y detectaban que era spam, lo reportaban y se listaba en la lista negra. Y durante un tiempo funcionó bastante bien, las usaba mucha gente porque era muy potente rechazar los correos no deseados antes de que estos se enviaran.

El inconveniente de este sistema es que necesita de 'falsos positivos'. Necesita que alguien intente enviar un correo legítimo, encuentre que no lo puede enviar, averigüe la razón (las listas negras), y lo comunique al administrador de su servidor de correo, forzándolo de esta manera a ponerle remedio. Es decir, que siempre se pierden correos. Además los administradores de las listas negras pasaron a tener mucho poder, y se han reportado muchos casos de abusos, donde los admins. añadían a gente a las listas negras por motivos injustificados [22] [23] (motivos personales, competencia desleal...). La fiebre por las listas negras creció desmesuradamente, y había listas que bloqueaban ISPs completos (por ejemplo, Telefónica ha estado en listas negras muchas veces [24]), y hasta países enteros (Corea, Nigeria), evitando cualquier correo que llegara de allí. El problema fue totalmente crítico cuando servidores de correo gratuitos, que son los que más usuarios tienen (Yahoo!, Hotmail, Lycos), fueron añadidos a las listas negras (quizás injustificadamente, o quizás porque se usaron para enviar spam). Estos servidores decidieron que ellos no eran responsables del spam, y que no tenían por qué hacer nada para salir de las listas, forzando a dejar de usar las listas negras a quien quisiera recibir sus correos. Y es lo que sucedió, perdiendo este método su eficacia.

3.3.- Filtro por firma

► Filtro por firma (checksum): Este método funciona tratando los correos no deseados como si fueran virus. Se busca una forma de encontrar una firma, un algoritmo que dé una serie que sea diferente para prácticamente cualquier correo (por ejemplo su checksum), y se guarda una lista de las firmas de los correos que son considerados SPAM. Cuando el mensaje llega al MTA, este calcula su firma y la busca en la lista de firmas SPAM. Si la encuentra, rechaza el mensaje. Si no la encuentra, lo procesa. Las implementaciones más destacables de este método son DCC (Distributed Checksum Clearinghouse) [25] y 'Vipul's Razor' [26].

El problema de este método es el mismo que el de los virus: necesitan que alguien reciba el spam y lo notifique. Y, además, desde que se recibe el primer correo no deseado, hasta que se encuentra la firma y se actualiza el filtro pueden pasar horas, y durante este tiempo el correo va entrando. Y los correos son mucho más abundantes y variados que los virus, por tanto este problema se intensifica.

Los spammers intentan pasar este filtro haciendo que los correos 'muten', es decir, que cambien su forma, aunque no su contenido. El mensaje de propaganda lo difunden igual pero le añaden palabras aleatorias al principio y/o al final, que hacen que se tarde más en poder definir la firma representativa.

3.4.- SpamPots - SpamTraps

► SpamPots - SpamTraps [27]: Para intentar detectar los correos no deseados antes que lleguen a un usuario se idearon los SpamPots o SpamTraps. La idea consiste en publicar direcciones de correo en sitios donde se sospecha que los spammers recogen direcciones de correo para llenar sus 'listas víctimas'. Estas direcciones no pertenecen a nadie, por tanto no deberían recibir ningún correo legítimo. Por tanto, todo el correo que se reciba será considerado spam, y se podrán crear las reglas adecuadas (firma/bayes) para que los filtros no lo dejen pasar.

Este método no tiene ningún filtro de correo de por sí, sólo permite aprender de los correos de spam de forma automática, sabiendo seguro que lo que se ha recibido es spam, pero se tienen que aplicar otros métodos para obtener resultados.

3.5.- Listas grises

► Listas grises [28]: Los spammers, habiendo visto que Internet se ha vaciado de casi todos los OPEN RELAYS, han tenido que montar sus propios servidores de correo para enviar el spam, y han buscado otras maneras de ahorrarse el máximo de tráfico posible, ahora que lo tienen que pagar ellos. Han recortado el protocolo SMTP, y han eliminado de los mensajes las cabeceras que han podido, y lo más importante, para acelerar el funcionamiento han hecho que durante el envío no se espere respuesta. Normalmente cuando un servidor de correo envía un correo, espera a que el servidor destino conteste si lo ha recibido correctamente o no, y en caso de no recibir confirmación, vuelve a intentar el envío. Los spammers, como son conscientes de que muchísimas de las direcciones de correo que usan como víctimas son incorrectas o inexistentes, han encontrado que el tráfico que dedicaban a los mensajes de error era demasiado alto, y, por tanto, la mayoría han optado por no esperar esta respuesta. Si el mensaje se envía bien al primer intento, perfecto. Pero si no lo hace, no es rentable analizar el error y actuar en consecuencia. Y es en este punto donde se fundamenta la idea de las listas grises [29]. La idea de las listas grises es mantener una lista de servidores 'familiares', de los cuales se aceptan los correos sin restricción, y hacer pasar una prueba a los 'no familiares', intentando distinguir entre la forma de actuar de un servidor de correo normal, que cumple los RFC del correo electrónico, de los servidores de correo 'recortados' de los spammers. Y habiendo comprobado que un servidor 'no familiar' cumple los RFC, se le añade a la lista de 'familiares'.

El funcionamiento es el siguiente: El servidor de correo mantiene una lista de tripletas 'familiares', que relacionan email remitente, IP remitente y dirección destino. Cuando llega un correo que tienen una tripleta que no está listada, el MTA le devuelve un 'Error Temporal', y se añade su tripleta a la lista. El RFC 2821 [30] dice que un servidor de correo que reciba un 'Error Temporal' tiene que reintentar el envío del correo pasado un rato. Si el correo lo había enviado un servidor de correo, reintentará al cabo de un rato, y entonces el correo se aceptará. Tendrá el inconveniente que este correo se retardará entre 5 y 30 minutos, pero a partir de ese momento, siempre que este remitente envíe un correo a este destinatario a través de esta IP de servidor, el correo se aceptará sin restricciones. En caso de que sea un servidor de correo 'recortado' (como los de los spammers), no se reintentará y por tanto no llegará nunca.

Este método es bastante nuevo, y no está muy extendido entre los servidores de correo, por tanto los spammers aún no han hecho acciones masivas para evitarlo. Por este motivo, aunque es muy sencillo y fácilmente evitable, es muy efectivo (aproximadamente el 90% del spam se descarta antes de su transmisión gracias a este método).

Las ventajas son su actual efectividad, y que se rechazan los correos spam antes de que sean transmitidos, ahorrando el tráfico que comportan.

Proxy caché - Squid

Como inconvenientes tenemos que siempre que un usuario recibe un mensaje legítimo de un remitente nuevo, el mensaje se retardará entre 5 y 30 minutos. Aunque si tenemos en cuenta que normalmente la gente suele recibir mensajes de los mismos remitentes, y los remitentes nuevos representan el 1% mensual, y además se trata de un retraso moderado, parece un inconveniente asumible.

Otro inconveniente es que cuando el método se adopte en muchos MTA, y los spammers encuentren en el un problema, no les será muy difícil pasar por encima. Algunas implementaciones de spammers han conseguido detectar los emails no distribuidos, reenviandolos inmediatamente, y es por eso que las implementaciones de las listas grises han forzado que desde el primer intento hasta el segundo tenga que pasar un tiempo prudencial (actualmente 5 minutos)

Evidentemente este sistema a la larga no es nada potente, porque tarde o temprano se neutralizará con una simple implementación correcta de los RFC. Pero si lo combinamos podría darse esta situación: si recibimos un email de spam, y lo rechazamos durante 30 minutos, ganamos tiempo para que el mismo mensaje pueda llegar a una spamtrap, que cree la firma y la añada al filtro de firmas. De esta forma, alomejor cuando el spammer vuelve a enviar el correo, nuestro servidor ya estará 'vacunado' contra este tipo de mensaje. Y en el peor de los casos, aunque esto no sucediera, habríamos incrementado el coste del envío del correo por parte del spammer. Habremos hecho un poco más difícil el envío de correo no deseado, haciendo que sea un poco menos rentable enviarlo. Habremos subido la copa del árbol unos centímetros.

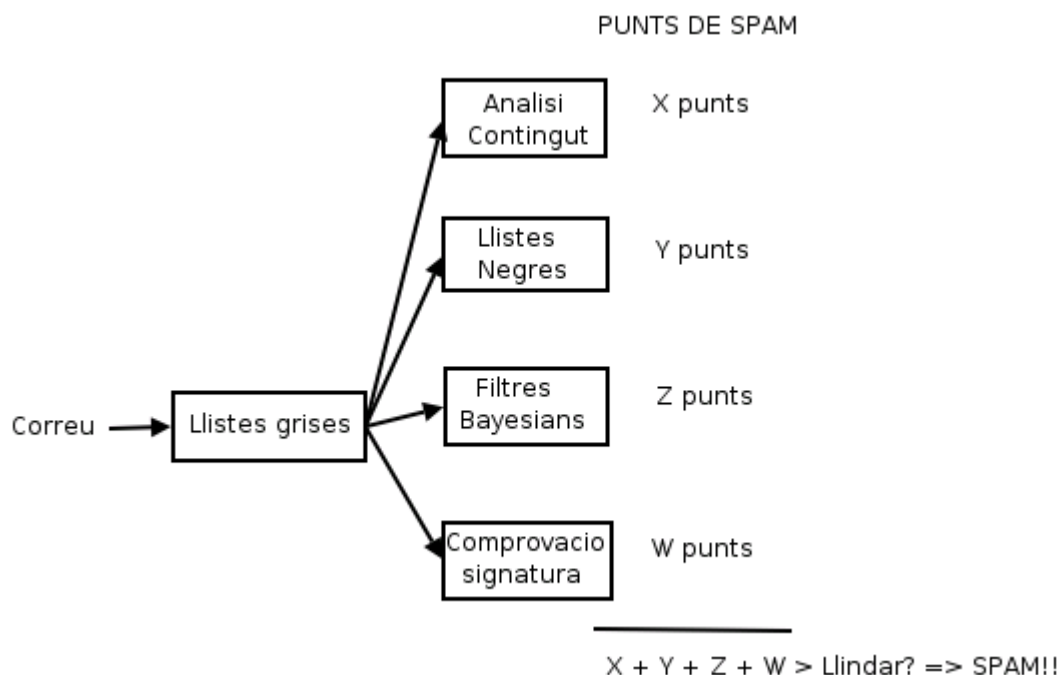
PUNTUANDO LOS DIFERENTES MÉTODOS DE DETECCIÓN

Como hemos podido comprobar, ninguno de los métodos anteriores funciona por sí sólo (con la excepción de las listas grises, que temporalmente tienen una eficacia muy elevada). Por tanto se tiene que buscar una forma de coger lo mejor de cada método.

Si nos fijamos, todos los métodos tienen unos puntos débiles diferentes. Es decir, un 'falso positivo' del análisis de contenido, muy probablemente no sea 'falso positivo' por firma, y un mensaje que sea bloqueado por una lista negra no tiene por qué ser bloqueado por análisis de contenido. Entonces si combinamos todos los métodos, podemos tener una herramienta realmente potente.

Valoramos qué métodos son más fiables que otros, y les damos puntuación según la fiabilidad que nos den. Decidimos una puntuación umbral a partir de la cual los mensajes son considerados SPAM. Entonces, por cada mensaje usamos TODOS los métodos, y sumamos los puntos de cada método que considere el mensaje como spam. Si la puntuación resultante supera el umbral definido, el mensaje se considera spam y se rechaza.

EJEMPLO: Llega un mensaje. Antes que nada tiene que pasar el filtro de las listas grises. Una vez lo haya pasado, primero se comprobará si el servidor remitente está listado en las listas negras, y se sumará puntos por cada lista donde esté listado. Después se analizará el mensaje según su contenido, y se sumarán puntos si aparecen según qué palabras (xanax, valium, pharmacy online...), si le faltan cabeceras, y si no pasa los filtros bayesianos... Después se consultará a Razor y a DCC. Y por cada positivo se sumarán puntos. Y si al final se supera el umbral definido, el mensaje se rechazará (ver figura).



REFERENCIAS

-
- [1] <http://www.maestrosdelweb.com/editorial/batallaspam>
 - [2] Basura en el correo - <http://www.baquia.com/com/legacy/9222.html>
 - [3] <http://www.paulgraham.com/antispam.html>
 - [4] <http://whitepapers.zdnet.co.uk/0,39025945,60085141p-39000628q,99.htm>
 - [5] <http://www.clickz.com/news/article.php/33115541>
 - [6] http://en.wikipedia.org/wiki/False_positive
 - [7] http://virusbusters.itcs.umich.edu/forged_spam.html
 - [8] <http://www.modwest.com/help/kb9-234.html>
 - [9] <http://members.cox.net/joejob>
 - [10] <http://www.joewein.de/sw/spam-joejob-info.htm>
 - [11] The Joe Job DoS attack - http://www.theregister.co.uk/2004/04/06/joejoe_dos_attack
 - [12] <http://www.templetons.com/brad/spam/spam25.html>
 - [13] <http://www.colorado.edu/its/email/spamfaq.html#5>
 - [14] <http://www.paulgraham.com/spam.html>
 - [15] http://www.hrc.es/bioest/Probabilidad_18.html
 - [16] <http://www.paulgraham.com/sofar.html>
 - [17] Random acts of spamness - http://wired-vig.wired.com/news/infostructure/0,1377,61886,00.html?tw=wn_tophead_2
 - [18] What is a black list? - <http://www.spam-blockers.com/SPAM-blacklists.htm>
 - [19] Open Relays - http://en.wikipedia.org/wiki/Open_mail_relay
 - [20] <http://www.spamhaus.org>
 - [21] <http://www.sconconsult.com/bill/dnsblhelp.html#3-8>
 - [22] <http://www.iadl.org/sorbs/sorbs-story.html>
 - [23] <http://www.iadl.org/sorbs/sorbs-email.html>
 - [24] Telefonica de España on spam blacklist - <http://www.webmasterworld.com/forum10/5344.htm>
 - [25] <http://www.rhyolite.com/anti-spam/dcc/>
 - [26] <http://razor.sourceforge.net/>
 - [27] <http://www.spamtrap.net.au/join/aboutspam/stoppingspam/spamtraps.php>
 - [28] <http://www.greylisting.org>
 - [29] <http://projects.puremagic.com/greylisting/whitepaper.html>
 - [30] <http://www.ietf.org/rfc/rfc2821.txt>