Engineering

Industrial & Management Engineering fields

Okayama University *Year* 2005

A protection method against massive error mails caused by sender spoofed spam mails

Nariyoshi Yamai
Okayama University

Kiyohiko Okayama
Okayama University

Takuya Miyashita
Okayama University

Shin Maruyama
Kyoto University

Motonori Nakamura
kyoto University

A Protection Method against Massive Error Mails Caused by Sender Spoofed Spam Mails

Nariyoshi Yamai[†] Kiyohiko Okayama[†] Takuya Miyashita[†]
Shin Maruyama[‡] Motonori Nakamura[‡]

[†]Okayama University
3-1-1, Tsushima-naka, Okayama 700-8530, Japan
{yamai,t_myst}@cc.okayama-u.ac.jp, okayama@cne.okayama-u.ac.jp
[‡]Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
marushin@net.ist.i.kyoto-u.ac.jp, motonori@media.kyoto-u.ac.jp

Abstract

Wide spread of spam mails is one of the most serious problems on e-mail environment. Particularly, spam mails with a spoofed sender address should not be left alone, since they make the mail server corresponding to the spoofed address be overloaded with massive error mails generated by the spam mails, and since they waste a lot of network and computer resources. In this paper, we propose a protection method of the mail server against such massive error mails. This method introduces an additional mail server that mainly deals with the error mails in order to reduce the load of the original mail server. This method also provide a function that refuses error mails to these two mail servers to save the network and computer resources.

1. Introduction

E-mail is one of the most popular services on the Internet as well as World Wide Web, and is an essential communication medium for social activities today. On the other hand, E-mail is one of the most troublesome services in terms of security. Particularly, the proliferation of spam mails (also referred to as Unsolicited Business E-mails or Unsolicited Commercial E-mails) is a serious issue of the Internet.

Spam mails cause extensive damage to the Internet community in various points of view as follows:

1. Users receiving many spam mails have to take much time for picking up valuable non-spam mails, or sometimes delete some non-spam mails by mistake.
2. The resources of mail servers (Mail Transfer Agent, MTA) and networks are wasted by the traffic of spam mails.
3. If the sender of spam mails is spoofed to an address of an existing domain, the spoofed sender and/or domain are misled to originate the spam mails.
4. If the sender of spam mails is spoofed to addresses of an existing domain, then huge volumes of spam mails sent to non-existing users are bouncing back to the MTA of the domain, and consequently the MTA is overloaded by massive error mails.

Among the above damages, the last one seems the most serious although it is not so frequent. For instance, on November 2002, an Internet Service Provider (ISP) in Japan received more than 300,000 error mails at a time, then mail delivery was delayed up to fifteen hours, and it took two and a half days for recovery to the normal operation. In addition, on October 2003, at least two domains in the United States had been received hundreds of thousands of error mails from all over the Internet[1].

Practically, a barrage of error mails bouncing back to the same spoofed sender domain caused by spam mails is a kind of Distributed Denial of Service (DDoS) attack. However, since most existing MTAs on the Internet allow users to send their mails without verification of the sender address, it is difficult to prevent malicious users from sending such spam mails.

With respect to this kind of DDoS attack, also known as “Joe job”, several documents have been issued so far[2, 3, 4, 5]. Stefan Frei, et al.[2] have analyzed the impact of this DDoS attack and show some recommendations such that “Do not accept mail for invalid recipients”, “Limit the

Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05)
0-7695-2262-9/05 $ 20.00 IEEE

maximum number of recipients", "Generate few error messages", and so on. These recommendations are, however, intended only to reduce traffic of bounce mails from each MTA, and effective only after most MTAs on the Internet follow them. Postfix[4] provides a method for blocking some kinds of error mails using header checking. However, this method does not reduce the load of the target MTAs since this method receives all error mails at first and then discards most of them. Thus, most existing methods seem ineffective for protecting normal mail delivery.

In this paper, we propose a protection method of normal mail delivery against such massive error mails. In order to reduce the load of the original MTA, this method introduces an additional MTA that mainly deals with the error mails. This method also provides a function that refuses error mails to these two MTAs to save the network and computer resources.

2. Error mails caused by sender spoofed spam mails

According to the fact that CD-ROMs containing more than one million e-mail addresses are advertised by spam mails, we simply suppose that at least one million spam mails of the same content are sent at a time. Since part of addresses in such CD-ROMs are invalid, many spam mails are to bounce back as error mails to the senders. For example, if a CD-ROM contains only 10 % of invalid addresses, more than 100,000 mails are to bounce back.

Meanwhile, since most existing MTAs on the Internet do not have a verification mechanism of the sender address, practically almost all spam mails are sent with a spoofed sender address so that the receivers do not find out the real sender. In this case, if a set of spam mails have the same spoofed sender address, all the error mails are sent to the MTA corresponding to this address (the victim MTA) at a time. Accordingly, the resources of the victim MTA such as CPU, memories and disks would be wasted by the traffic of massive error mails, and then the mail delivery would be delayed or the MTA would go down in the worst case, regardless of whether the spoofed sender address is existing on the victim MTA or not.

Error mails are sent from two kinds of sources.

One kind is the MTA originating spam mails (the originating MTA). If a spammer originates spam mails with the originating MTA, it tries to deliver spam mails to other MTAs corresponding to their destinations (the destination MTAs). Then, if a destination MTA refuses to receive the mail due to "User unknown" or "Host unknown", the originating MTA generates an error mail and tries to send it to the spoofed sender (see Figure 1). In the rest of the paper, we call this type of error mails *direct error mails*.

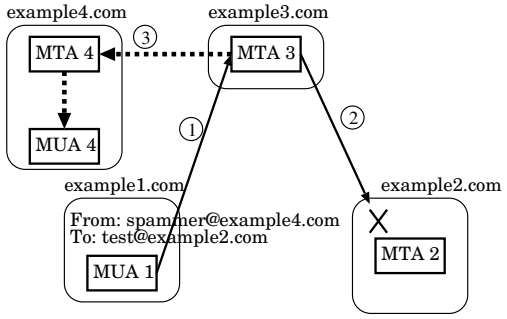


Figure 1. Delivery of error mails (1)

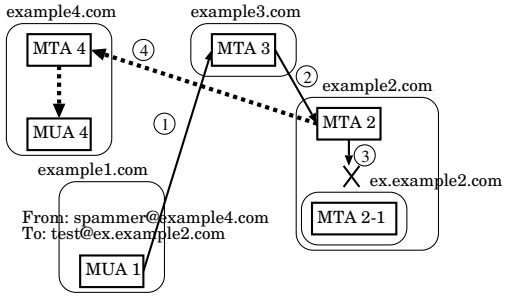


Figure 2. Delivery of error mails (2)

The other kind is a mail gateway for a destination address. If there exists a mail gateway in front of a destination MTA, the mail gateway receives the spam mail and tries to forward it to the destination MTA. Then, if the destination MTA refuses to receive the mail, the mail gateway generates an error mail and tries to send it to the spoofed sender (see Figure 2). In the rest of the paper, we call this type of error mails *redirected error mails*.

In order to protect the victim MTA against massive error mails as mentioned above, we have to cope with both kinds, chiefly the latter kind of error mails. However, with existing methods, it is difficult to protect the victim MTA against redirected error mails since there exist so many mail gateways on the Internet that are potential senders of error mails and innocent. For example, filtering based on IP address or rate control for SMTP traffic, which are usual protection techniques against DoS attacks, are not suitable for this problem since they restrict delivery of not only error mails but also normal mails. Load sharing with additional MTAs is another possible solution. However, on the above mentioned incident on November 2002, although four MTAs were employed, it took no less than two and a half days for recovery.

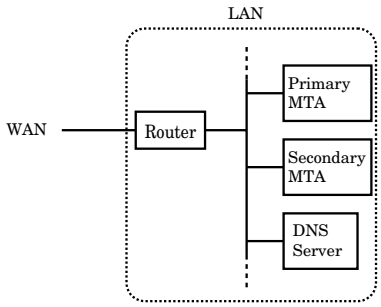


Figure 3. Environment of the proposed method

3. Protection against Massive Error Mails

3.1. Outline of the proposed method

As mentioned in the previous section, load sharing among MTAs is a possible solution of this kind of problems. However, if all MTAs have the same role, all of them would receive still so many error mails and consequently normal mail delivery would be affected.

In order to protect normal mail delivery, we propose a method that introduces another (secondary) MTA mainly dedicated for error mails, whereas the original (primary) MTA is mainly dedicated for normal mails, on the environment shown as Figure 3. With this method, while the secondary MTA would heavily loaded, the primary MTA would receive normal mails as usual.

On this method, the most important issue is how to divert only error mails to the secondary MTA. To solve this issue, we introduce filtering on the ingress router and manipulation of MX records of the DNS servers on the environment shown in Figure 3.

3.2. Protection against direct error mails

Direct error mails are likely to be a major part of error mails, but are relatively easy to be diverted to the secondary MTA.

First of all, the administrator configures the DNS server of the destination domain to have the secondary MX record referring to the secondary MTA in advance. If an MTA outside of the domain is sending many error mails to the primary MTA, then the router is configured to filter out the SMTP connection between the originating MTA and the primary MTA. After this configuration, since the originating MTA fails to send error mails directly to the primary MTA, it becomes to send error mails to the secondary MTA, according to the secondary MX record (see Figure 4).

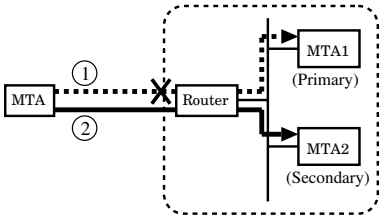


Figure 4. Delivery of direct error mails

Note that this method requires the IP address of the originating MTA, which is easily obtained from the SMTP log of the primary MTA.

3.3. Protection against redirected error mails

In contrast to direct error mails, redirected error mails are sent from many mail gateways, each of which sends only a few error mails. Therefore filtering out the SMTP connection from such mail gateways at the router does not seem effective because still less error mails from each mail gateway are likely to be filtered out after introducing each filtering rule. In addition, it requires so many filtering rules that degrade the performance of the router.

Alternatively, such mail gateways are likely to resolve the MX record of the destination domain at the moment they try to send error mails, provided that they have never sent any mails to that domain recently. On the other hand, MTAs frequently sending normal mails do not have to resolve the MX record usually since they already have the result in their cache.

Utilizing this property, we propose a method that deletes the primary MX record at the beginning massive error mails are detected on the primary MTA. In addition to deletion, the proposed method also changes the Time To Live (TTL) values of the MX records. Specifically, on the normal condition, TTLs of the primary and secondary MXs are set to a large value (one week for example), whereas after deletion, TTL of the secondary MX are set to a small value (one hour for example).

With this method, mail gateways without the MX record in their cache send error mails to the secondary MTA as shown in Figure 5, whereas MTAs frequently sending normal mails send normal mails to the primary MTA using the cached MX record as shown in Figure 6. Consequently, the proposed method can protect the primary MTA against massive error mails, along with minimal impact on normal mail delivery.

Note that, in this method, the secondary MTA have to be configured to forward normal mails other than error mails to the primary MTA, regardless of the MX record setting.

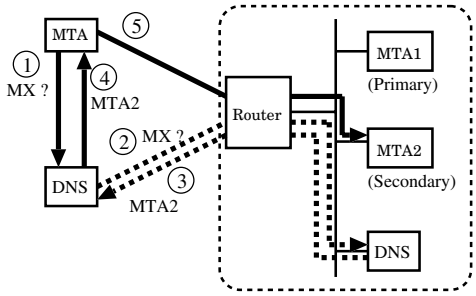


Figure 5. Delivery of redirected error mails (1)

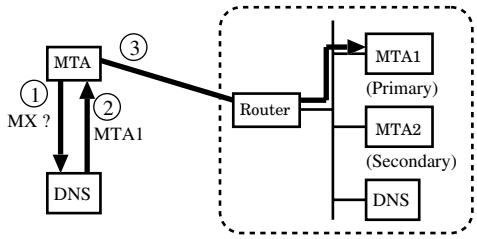


Figure 6. Delivery of redirected error mails (2)

3.4. Error Mail Handling

In the proposed method, since the secondary MTA may receive some normal mails as well as most error mails, it is important for the secondary MTA to deal with error mails as fast as possible in order to keep delivery delay of normal mails short. Therefore, the secondary MTA should reject or discard an error mail before it receives its entire body, while accepting and forwarding any other mails. In the proposed method, MTAs deal with mails for the specific addresses (possible spoofed sender addresses) as follows:

1. If a mail has an empty Envelope-From, namely <>, it would be rejected immediately after receiving "RCPT TO: <*spoofed sender address*>" command.
2. Otherwise, the entire body of the mail is received. Then the accepting MTA checks the From: header of the received mail. If it is "MAILER-DAEMON" at some domain, the MTA logs the fact and discards this mail.
3. Otherwise, the received mail is delivered to the recipient as a normal mail.

With this process, we can reduce the overheads of dealing with error mails. Note that, based on this process, we can automatically reply against complaint error mails, for example by checking whether the mail sent to the spoofed sender address includes the original spam mail or not.

3.5. Invocation and termination

Generally, in order to make the proposed method effective, it is important to detect a symptom of error mail rush as early as possible. As the indices of the symptom, the following two events are effective[6]:

1. When the primary MTA receives as many as a given number of error mails destined for the same address in a given period.
2. When the DNS server receives as many as a given number of queries for the same MX record in a given period.

These two types of events may occur during normal operation. For example, as for the former event, when a mailing list managed on an MTA sends a message to many subscribers, some of them are sometimes bouncing back to the MTA and it detects the error mails as a symptom of error mail rush by mistake. As for the latter event, many normal mails are accidentally delivered to a MTA in a short period. However, even if the protection operation invokes accidentally, normal mails are properly delivered to the primary MTA via the secondary MTA.

On the other hand, the protection operation terminates when error mails destined for the spoofed address are not received on the secondary MTA for a given period. Note that error mails received on the primary MTA are not used for decision on the termination, since they are sent from the MTA having the primary MTA in its cache as the primary MX and are not affected by the protection operation.

4. Design of the prototype system

4.1. System Configuration

Since many domains have two or more DNS servers and secondary MTAs, the proposed method should be carefully designed so that these DNS servers and MTAs cooperate with one another. Particularly, the invocation and the termination of the protection operation have to be synchronized on all components.

In the prototype system, the primary DNS server is designed to be charged with the master controller of the system, as shown in Figure 7. Other components, namely secondary DNS servers, the primary MTA and secondary MTAs, have a slave controller, communicating with the master controller on the primary DNS server.

A controller on a secondary DNS server monitors the query log to see whether as many as the given number (called *the notification threshold value*) of queries for the

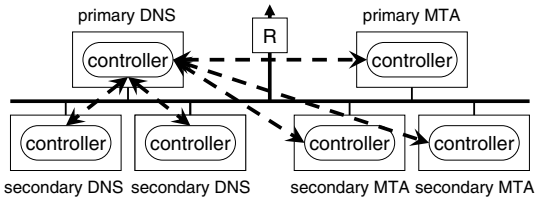


Figure 7. Configuration of the prototype system

same MX record are received in a given period or not. If so many queries are monitored, the controller notifies this fact to the master controller. The master controller has another threshold value called *the invocation threshold value* for invocation of the protection operation. Specifically, if the master controller confirms as many queries for the same MX record as the invocation threshold value, including those received on the primary MTA itself and those notified by secondary MTAs, it invokes the protection operation.

The controller on the primary MTA monitors the MTA log to see whether as many as the given number of error mails for the same address are received in a given period or not. If so many error mails are received, the controller notifies this fact to the master controller. On receiving the notification, the master controller immediately invokes the protection operation.

On the other hand, the slave controllers on secondary MTAs monitors the MTA log to see whether no error mails for the same address are received in a given period or not. If no error mails are received, the controller notifies this fact to the master controller. After receiving the notification from all secondary MTAs, the master controller terminates the protection operation.

Note that this configuration allows additional secondary DNS servers and secondary MTAs to be introduced easily, only by communicating with the master controller.

4.2. Overall Procedure

We summarize the overall procedure of the proposed method as follows:

1. As an initialization phase, the primary MTA and secondary MTAs are registered on all DNS servers as the primary MX and secondary MXs. In addition, TTLs of these MXs are set to a large value.
2. The master controller and slave controllers monitor their logs to detect a symptom of error mail rush described in the previous section. If any of slave controllers detect the symptom, it notifies the fact to the master controller.
3. When the master controller decides to invoke the protection operation, it removes the primary MX records on the primary and all secondary DNS servers. In addition, TTLs of secondary MX records are set to a small value.
4. If a slave controller on an MTA detects the originating MTA, it notify the IP address of the originating MTA to the master controller. The master controller configures the router to filter out any SMTP connection from the originating MTA to the primary MTA and notifies the IP address of the originating MTA. The secondary MTAs then marks any mails from the notified originating MTA. If a controller on an MTA detects many error mails destined for a specific address (possible spoofed sender address), it notifies the destination address to the master controller on the primary DNS. The master controller then notifies the address to controllers on all MTAs. Each controller on MTAs then updates the configuration of the MTA to reject error mails destined for the notified address.
5. The master controller continues to monitor the log to see whether many error mails described in section 3.5 are received or not. If so, go back to step 4. Slave controllers on secondary MTAs monitor their logs and perform the following process:
   - If a slave controller does not detect error mails destined for a spoofed sender address for a given period, it sends a notification message to the master controller. After receiving notification messages from all secondary MTAs, the master controller sends a cancellation message about this address to all MTAs. After that, each MTA cancels the special processing for this address.
   - If a slave controller does not detect error mails from an originating MTA for a given period, it sends another kind of notification message to the master controller. After receiving notification messages from all secondary MTAs, the master controller removes the filter of the router against the originating MTA and sends a cancellation message about this originating MTA to all secondary MTAs. After that, each secondary MTA cancels the special processing for this originating MTA.
   - If a slave controller does not detect any kind of error mails that are the targets of monitoring for a given period, it sends a termination message to the master controller. After receiving termination messages from all secondary MTAs, proceed to the next step.

6. The master controller restores the primary MX records on all DNS servers. In addition, TTLs of all MX records are reset to a large value. Then go back to step 2.

5. Implementation and verification of the prototype system

5.1. Implementation of the prototype system

Based on the design discussed in the previous section, we have implemented a prototype system. In the prototype system, we introduced two secondary MTAs and two secondary DNS servers in addition to the primary MTA and the primary DNS server, and all of them were allocated to different PCs running FreeBSD (version 4.5 or 4.9). Since a filtering function provided by IP firewall feature of FreeBSD is available on the primary MTA, the router was omitted from the prototype system.

We used BIND version 9.2.2[7] for all DNS servers. On the DNS servers, dynamic DNS update feature was used for updating MX records, and TSIG-signed update feature was used for rejecting abusive update requests from the outside of the prototype system. To propagate the update of MX records on the primary DNS server to secondary DNS servers immediately, DNS NOTIFY feature was also engaged. On the other hand, we used sendmail version 8.12.9[8] for all MTAs. To monitor and control the logs of DNS servers and MTAs, we wrote a Perl script and ran on every server.

5.2. Simulation experiments

In order to verify the behavior of the prototype system, it is desirable to send many sender spoofed spam mails to the Internet in practice, but this method has moral hazard. Therefore, we constructed the experimental network isolated from the Internet and made several simulation experiments on this network.

Firstly, we examined the behavior of the DNS servers by sending several queries of an MX record from outside of the prototype system to the DNS servers. In this experiment, the invocation and notification threshold value were set to 4 queries and 2 queries per 10 minutes, respectively, according to the results of the preliminary experiments[6]. As the result of this experiment, when queries was exceeded either the invocation threshold on one DNS server or the notification threshold on more than one DNS servers, the prototype system detected the symptom correctly, and then MX records of the primary MTA were deleted on the primary and all secondary DNS servers.

Secondly, we examined the behavior of the MTAs by sending massive error mails from an external MTA. In this experiment, the protection operation was configured to be started when an MTA receives more than 9 error mails per 10 minutes from the same MTA, according to the results of the preliminary experiments[6]. As the result, while the external MTA sent a thousand error mails, the primary MTA and each of secondary MTAs received only 10 and 4 error mails respectively, and remaining error mails were rejected.

Finally, we examined the termination of the protection operation. In this experiment, the prototype system was configured to reset to the initial state if all MTAs receive no error mails for more than 10 minutes. As the result, the prototype system reset to the initial state at the 10 minutes after the last error mail had been received.

According to these results, the prototype system works correctly as we expect on these experiments.

Number of all mails:	73,300
Number of error mails:	73,086
Number of sending MTAs:	22,272

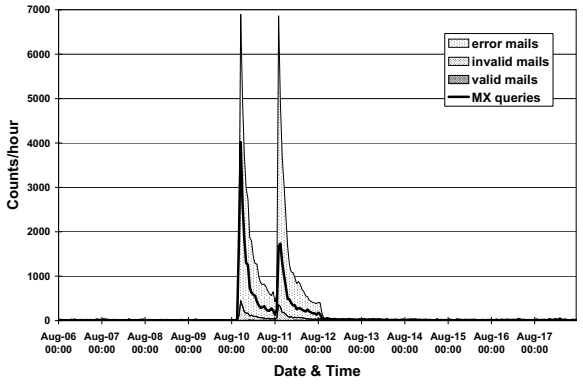
Table 1. Statistics of mails received between Aug 10 4:00 – Aug 12 3:00

5.3. Verification of effectiveness

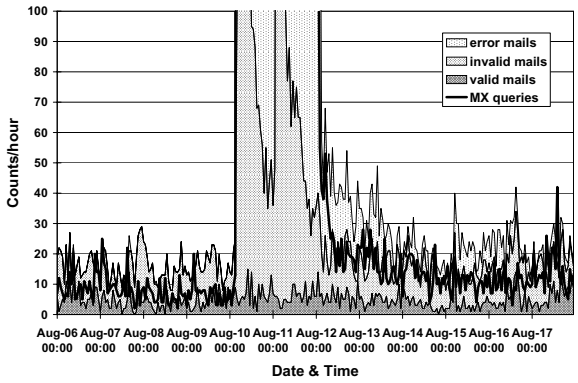
Generally, it is difficult to verify the effectiveness of the proposed method since we could not send spam mails with an address of our domain as the sender in practice by ethical reason. Therefore, we were looking forward to an attack of massive error mails to our domain for a long time. Consequently a subdomain of Okayama University got attacked by more than 70,000 error mails from August 10 at 4 a.m. through August 12 at 3 a.m.

Since the prototype system were not deployed during the attack, instead we analyzed the logs of the mail gateways and those of DNS servers.

The statistics of mails received during the attack are shown in Table 1. The numbers per hour of mails with the null sender address (error mails), accepted mails other than error mails (valid mails), and rejected mails other than error mails (invalid mails) are shown as stacked area graph in Figure 8. The number of queries for the MX record of the victim domain is also shown in this figure as a line graph. According to Figure 8, this attack consisted of two phases. In both phases, the more error mails were sent to the mail gateways, the more MX queries were recorded. This shows that many MTAs sending error mails do not have the MX record in their cache as we expected. Note that many invalid mails were also sent to the mail gateways during the attack. Most of these invalid mails were warning messages generated by



(a) overall view



(b) magnified view

Figure 8. The numbers per hour of mails and MX queries

anti-spam systems, automatic replies telling the absence or a forwarding address and so on.

Then, we tried to estimate the effectiveness of the proposed method. By analysis of the logs of the mail gateways, the number of error mails sent to the primary MTA after the beginning of the attack would be 209 (0.3%) out of 73,086 if the proposed method were deployed. In fact, these error mails were sent from the MTAs that had ever sent some mails to the mail gateways before the beginning of the attack. On the other hand, the number of valid mails except spam mails sent to the secondary MTA would be 29 (16%) out of 187. In fact, most of the valid mails were sent from an MTA dealing with some mailing lists that might happen to expire and query the MX record. The rest of valid mails were sent from another MTA that might invalidate all the cached record once per day. This value seems somewhat large but we can decrease the number of such valid mails by introducing *DNS whitelist*[9].

According to the above analysis, we summarize that the proposed method can fairly separate massive error mails from normal mails and can effectively protect the primary MTA against massive error mails.

6. Conclusions

In this paper, we described the design and implementation of a method reducing the impact of massive error mails generated by sender spoofed spam mails against the normal mail delivery. In addition, we have confirmed that the proposed method would work well with respect to the attack on a subdomain in Okayama University. Further works include the evaluation of the proposed method on other domains and the investigation of several parameters, such as criteria for invocation or termination of the protection operation.

Acknowledgment

This research was partially supported by Japan Society for the Promotion of Science(JSPS), Grant-in-Aid for Scientific Research (C)(2) No. 15500039 in 2003–2004.

References

[1] Brian McWilliams: "Wired News: Time-Travel Spammer Strikes Back", Lycos, Inc., `http://www.wired.com/news/technology/0,1282,61026,00.html`, November 2003.

[2] Stefan Frei, Ivo Silvestri, and Gunter Ollmann: "Mail Non-Delivery Notice Attacks", `http://www.techzoom.net/mailbomb`, April 2004.

[3] "Mail Non-Delivery Attack", Alert Number: AL04-005, Public Safety and Emergency Preparedness Canada (PSEPC), `http://www.ocipep-bpiepc.gc.ca/opsprods/alerts/2004/AL04-005_e.asp`, April 2004.

[4] "Postfix Backscatter Howto", `http://www.postfix.org/BACKSCATTER_README.html`.

[5] "SpamCop.net - SpamCop FAQ: How can I control unsolicited bounces?", `http://www.spamcop.net/fom-serve/cache/380.html`, 2004.

[6] Kiyoshi Tanaka, Nariyoshi Yamai, Kiyohiko Okayama, Takuya Miyashita, Motonori Nakamura, and Shin Maruyama: "An Early Detection Method of Denial of Service Attack caused by Sender Spoofed Spam Mails", Proceedings of the 64th IPSJ General Conference, 2H-2, March 2002 (in Japanese).

[7] Internet Systems Consortium Inc.: "ISC BIND", `http://www.isc.org/sw/bind/`, 2004.

[8] Sendmail, Inc.: "Sendmail Home Page", `http://www.sendmail.org/`.

[9] Shin Maruyama, Motonori Nakamura, Yasuo Okabe, and Nariyoshi Yamai: "A Dynamic Modification on DNS Response and its Application on Mail Transfer Agent", IPSJ SIG Technical Report, 2004-DSM-32-14, pp.79–84, March 2004 (in Japanese).