23

Spam Mail Blocking in Mailing Lists

Kenichi Takahashi[1], Akihiro Sakai[12] and Kouichi Sakurai[12]
[1]*Institute of Systems, Information Technologies and Nanotechnologies*
[2]*Faculty of Information Science and Electrical Engineering, Kyushu University*
Japan

1. Introduction

The increasing popularity of the Internet has led to e-mail becoming one of the widely used essential services in our personal and business life. However, in the recent times, the number of spam mails received has increased rapidly. Symantec has reported that spam mails account for up to 75% of all e-mails (Symantec, 2008). Such a large amount of spam mails waste the valuable time of e-mail users who have to filter out these spam mails. Moreover, the large number of spam mails may prevent users from seeing important mails. Some spam mails may contain viruses, worms, or links to a phishing site and cause information leakage, loss of information, invasion of privacy, and damage to computers.

Therefore, many researchers have proposed and elaborated on several techniques for dealing with spam mail. Spam filtering is one of the widely used techniques against spam mails (Tabata, 2006). Spam filters infer whether a mail is a spam mail or not on the basis of certain preserved keywords in the mail. However, spam filtering techniques also give rise to false positive and false negative results. For example, a large number of spam mails include the word "Viagra." Therefore, a mail including Viagra will usually be assumed to be spam. However, users belonging to a drug company may usually use Viagra in their mails. In such cases, spam filters may produce false positive results. Moreover, AT&T's anti-anti-spam patent (Pfleeger and Bloom, 2005) reveals that the current spam filtering techniques are actually a cat-and-mouse game with spammers.

Whitelisting/blacklisting is one of the techniques used for blocking spam mails. An administrator notes down accepted/rejected mail senders and/or domains in a whitelist/blacklist. Then, mails from non-accepted/rejected mail senders and domains are blocked. However, we cannot expect to know everyone who would e-mail. Hence, it is difficult to specify mails from which mail senders and/or domains can be accepted/rejected. In this situation, we forgo blocking spam mails for our personal mails, but we focus on blocking spam mails in a mailing list. A large number of spam mails also come through in mailing lists. Mailing lists are used to disseminate information within specific groups such as laboratory staff, a project group, or users sharing the same interests. The members of a mailing list share a common mailing list address. When a member sends a mail to the mailing list address, the mail is automatically forwarded to all the mailing list members. Since the members of a mailing list share a common mailing list address, the probability of address leakage increases with an increase in the number of members. Meanwhile, the

mailing list address is shared only among mailing list members and should be used only for mailing to the mailing list. This means that the mailing list address should not be used for online shopping, user registration, etc. Thus, we can assume that the mailing list address is used only for posting mails to mailing list. Hence, changing of a mailing list address is easier than changing personal mail addresses because the influence of the change is limited to the mailing list members. However, frequent changes to the mailing list address are not acceptable since such changes would affect all of the mailing list members.

In this chapter, we introduce a system to block spam mails in a mailing list. In this system, a mailing list system assigns different posting mail addresses to different mailing list members. A mailing list member sends a mail to the posting mail address assigned to him/her in order to send a mail to the mailing list. When a spam mail is received, the posting mail address leading to the receiving of the spam mail is identified and invalidated, and a new posting mail address is assigned to the member. Thus, we can block the spam mails coming from the invalidated address, but the member can post a mail to the mailing list from the new posting mail address. Furthermore, our system is highly compatible with the typical mail systems because our system does not require any particular software to be installed on the client machines.

The remainder of this chapter is organized as follows. The next section analyzes the causes that lead to the receiving of spam mails. Section 3 introduces some techniques against spam mails such as spam filtering, blacklisting/whitelisting and so on. Section 4 introduces our mailing list system for blocking spam mails, and section 5 presents an evaluation of this system. Finally, in section 6, the chapter is concluded.

2. Causes of spam mails

We analyze the causes of spam mails in mailing lists. Generally, we have to know recipient addresses to send an e-mail; thus, spammers also have to obtain recipient addresses for sending spam mails. However, a mailing list addresses is necessary to be shared only by the mailing list members; hence, it should remain unknown to users (e.g. spammers) other than the mailing list members. Therefore, ideally, spammers should not be able to send spam mails to mailing list addresses, as these addresses are unknown to them. However, in fact, a large number of spam mails come through mailing lists. This is because of the leakage of the mailing list addresses. We can classify the causes of address leakage into the following cases (Figure 1).

Fig. 1. Causes of address leakage

1. A member puts the mailing list address on his Web pages. Then, a Web crawler collects the address. Alternatively, a member uses the mailing list address for online shopping user registration, etc.
2. A member sends an e-mail to both the mailing list and a non-member of the mailing list. Then, the non-member leaks the mailing list address.
3. A mailing list member uses a machine that has spyware installed. The spyware collects the mailing list address and leaks it to spammers.
4. There is an eavesdropper in a channel between a member and the mailing list server. Alternatively, when a member posts an e-mail to a mailing list from an un-trusted mail server, the un-trusted mail server leaks the mailing list address.
5. An attacker sends a mail to random addresses, and a mailing list address is unfortunately included in the addresses. Then, the attacker remembers the address as a valid address (DHA: Directory Harvest Attack).

In all of the abovementioned cases except for case 5, mailing list address leakage is caused by a mailing list member. A spammer collects a mailing list address leaked by these causes and sends spam mails. Once the address is leaked to a spammer, the address is exchanged between spammers. This leads to an increase in the number of spam mails. Thus, it has become impossible to stop spam mails.

3. Techniques against spam mails

3.1 Spam filtering

Spam filtering techniques are based on the differences in the characteristics of spam mails (Tabata, 2006) and legitimate mails. Some of these techniques use statistical classification based on machine learning. Most of them are Bayesian-based filters. These techniques learn and maintain words that are frequently used in spam mails and legitimate mails. Then, if a mail contains many words used in spam mails, the mail is judged as a spam mail. Further, there is a research on classifying spam mails according to the routing information (Fujii, 2004). However, these techniques produce false positive and false negative results.

3.2 Whitelisting and blacklisting

Whitelists and blacklists are an effective tool for blocking spam mails. They work well in the case of a mailing lists with a limitted posting methods, such as the mailing list of a magazine. Otherwise, it is difficult to list all the acceptable senders and/or domains because the administrator may not know which mail servers are used by the members. For example, it is difficult for a system administrator to list all the temporary methods that a user may use to post a mail, such as Gmail, Yahoo! Mail, cell phones or alternate SMTP servers. Otherwise, the usability will decrease because the administrator has to refuse these methods.

DNSBL (DNS-based Blackhole List) (dnsbl, 2008) provides the list of IP addresses which spam mails are sent frequently. For example, the Spamhaus Project (http://www.spamhaus.org/) provides such a list. Hence, we can filter spam mails according to the list. However, DNSBL is not precise; even legitimate mail servers such as those belonging to universities and government agencies are sometimes registered in this list. Then, the system filters out even legitimate mails. Moreover, DNSBL is not compatible with dynamic IP address systems.

3.3 Sender ID framework

Sender ID Framework (SIDF) (SIDF, 2008; Wong & Schlitt, 2006) is one of whitelisting approach. The overview of SIDF is shown in figure 2. In SIDF, the sender's domain appends *SPF records* in its DNS server. An SPF record contains the list of IP addresses and hostnames that are permitted to send mails from this domain. Then, receivers can check whether a mail is permitted or not by the sender's domain by the following steps:

1. Extracting sender's domain information from mail headers
2. Obtaining SPF records from the sender's domain
3. Checking whether the sender's IP address is listed in the SPF records

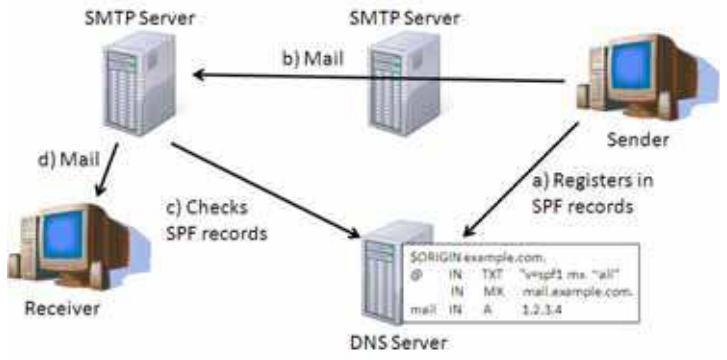

Fig. 2. Overview of Sender ID Framework

Thus, if the IP address is not listed in SPF records, the receiver judges the mail is spam because the SMTP server does not allow a mail from the IP address. However, this technique requires SPF records registered in the DNS server, installation of specific libraries to check the SPF records, and the cooperation of the Internet service provider. Moreover, mail forwarding used in mailing lists may result in false positives because the forwarding server is not listed in the sender's SPF records.

3.4 DomainKeys Identified Mail

DomainKeys Identified Mail (DKIM) (DKIM, 2008; Allman et al., 2007) is an approach to verify a sender's address on the basis of a digital signature. In DKIM, a public key to verify the sender's digital signature is prepared in the DNS server of the sender's domain. When a sender sends a mail, the sender's SMTP server automatically appends a digital signature to the mail header. Then, a receiver verifies the digital signature by using the public key on the sender's DNS server (figure 3). The following steps are carried out by the receiver to know whether the mail is spam or not.

1. Extracting the sender's domain information and digital signature
2. Obtaining a public key of the sender
3. Verifying the digital signature by the public key

Thus, if the verification is passed, the receiver is convinced that the mail is sent from a legitimate sender. However, this technique has the same problems as SIDF; it requires a list of public keys registered in the DNS server, installation of specific libraries to verify digital signatures, and has a mail fowarding problem.

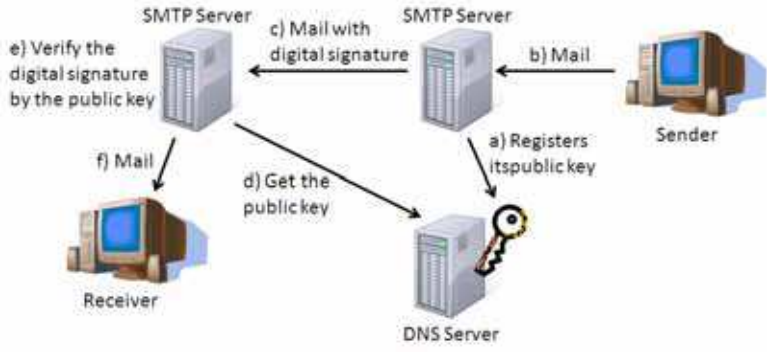

Fig. 3. Overview of DomainKeys Identified Mail

3.5 Other techniques

We can use disposal e-mail address (DEA) (Seigneur & Jensen, 2003) against spam mails. The representative DEA service providers are Spamex (http://www.spamex.com/) and myTrashMail.com (http://mytrashmail.com/). It is, however, not compatible with mailing list systems.

Spammers usually send a large number of spam mails. Therefore, some researchers have attempted a approach where a task is imposed on the mail sender (Dwork & Naor, 1993; Roman et al., 2005). In such an approach, when a receiver receives an e-mail from a sender, the receiver sends an some easy question back to the sender. If the sender does not give the correct answer, the receiver judges the mail as spam. The approach prevents spammers from sending a large number of spam mails because the computation power of the spammers to make the answer is limitted. (Kraut et al., 2005) and (Kuipers et al., 2005) have proposed charging mail senders some money instead of having a computation task. However, these approaches are not applicable to mailing list systems. In addition, the load of spam senders is also imposed on the legitimate e-mail senders. Moreover, we would then have to replace the current mail systems with their new versions, which seems to be impossible.

Privango (Takahashi et al., 2005) uses an encrypted condition as a mail address. When a user receives a mail that does not match the condition in the mail address, the system automatically rejects it. However, it is difficult for us to remember the mail address because the mail address is like a random string sequence.

4. Personal mailing list address to block spam mails

We propose a system to block spam mails in a mailing list; in this system, we assign different addresses to different mailing list members.

4.1 Requirements

Various techniques have been employed to block spam mails, such as SIDF and DKIM. However, these techniques require the cooperation of Internet service providers and the installation of particular software on the client machines. Therefore, these techniques cannot be easily applied to the current mail systems. We need to develop a technique that is

compatible with the typical mail systems. Thus, a system should satisfy the following requirements for compatibility with the typical mail systems:

- Users need not install particular software. Thus, they should be able to use their customized e-mail client systems.
- The technique should be compatible with the current standard mail protocols such as POP and SMTP.
- It should be easy to use with the typical mailing list systems. Thus, the following requirements should be satisfied:
  - Users can make use of an address, which is easy to remember, for mailing to a mailing list;
  - Users can mail to a mailing list from any mail address and any server, such as Gmail, Yahoo! Mail, and their own home mail servers;
  - Users can reply to a mailing list by using the same methods as those used in the typical mailing list systems.

Our system tries to block spam mails by invalidating the address which leads to receiving spam mails. Therefore, the system needs to identify such addresses. However, the identification of the member causing spam mails may be disadvantaged. Moreover, address leakage caused by DHA happens without any relation to members' carelessness. Therefore, we require the following:

- When a spam mail is received, the mailing list members should not be able to identify the member whose address led to the receiving the spam mails.
- The system should be able to distinguish an address leakage caused by DHA from other cases.

We have to develop a system that can block spam mails and also satisfy the abovementioned requirements.

4.2 Proposed system

In the typical mailing list systems, all mailing list members use the same mailing list address. This makes it difficult to change the mailing list address. Therefore, we assign a different address, named *individual address*, to each mailing list member. Then, each member sends mails to his/her assigned individual address. Thus, the individual addresses can be easily changed because each member uses a different individual address. The overview of our proposed system is shown in figure 4.

Fig. 4. Overview of proposed system

1. The administrator decides the mailing list members.
2. Then, the random individual addresses are created and assigned to each member. Here, each member can change his/her individual address to his/her customized address by using the additional steps.
3. A member sends a mail to his/her individual address.
4. When spam mails increase, the individual address that leads to the receiving of the spam mails is invalidated and a new individual address is assigned.

In this approach, each member uses a different individual address. Therefore, the invalidation of an individual address does not affect the other members. In other words, the system can block spam mails by changing the address of only the member causing the spam mails.

4.2.1 Creation of mailing list

A mailing list is created by an administrator. First, the administrator decides the address of the mailing list, named *ML address*, the members and the configuration of the mailing list. Note that the ML address is not used for the sending mails to the mailing list. The ML address may be used only in the To field of mails sent from the mailing list. The system ignores direct mails to the ML address. The configuration defines which address should be used in To, Reply-To and From fields, and a penalty to a member who leads to the receiving of spam mails. Then, the system creates an *Address-ML* table, an *Address-Sender* table, a *Penalty* table, and a *History DB*. The Address-ML table records the individual addresses along with the ML address. The Address-Sender table records the individual address and the mail address of a member to whom the individual address is assigned. The Penalty table manages the penalty values of each member, which increases when a spam mail is received. The History DB is a database that manages all the mails sent to the mailing list.

Next, random individual addresses are created, and these are recorded in the Address-ML and the Address-Sender table. Then, an invitation mail is sent out to each member.

When a member receives the invitation mail, he/she can send a reply to the mail including his/her customized address. If the customized address is already used by other member, an acceptance fail mail is sent to the member. Then, the member can try his next customized address. If the attempt is successful, the Address-ML and Address-Sender table are overwritten by the customized address (this will be his/her individual address) and an acceptance mail is sent to the member. As a result, the user can use his customized address, which the user can easily remember, for sending a mail to the mailing list.

Similarly, we can add other members after the mailing list is created.

4.2.2 Sending a mail to a mailing list

A member can send a mail to a mailing list by sending a mail to his/her individual address. The From, To and Reply-To fields of a mail forwarded to the mailing list members are defined by the configuration of each mailing list. Here, we assume that a ML address, a sender's address, and an address for replying to the mailing list are specified in To, From, and Reply-To fields, respectively (Figure 5).

When a mail is received by the mailing list, a pair of the mail and its Message-Id is recorded in the History DB. Next, the ML address corresponding to the To field (sender's individual address) of the mail is determined by the Address-ML table. Then, the To field is changed to

the ML address. After that, if and only if the mail has a Reply-To field, the From field is changed to the address of the Reply-To field. Finally, mails that are forwarded to each mailing list members are created by changing the Reply-To fields to the individual addresses corresponding to each member by using the Address-Sender table. Then, the mails are forwarded to the mailing list members.

Fig. 5. Flow of sending mail to mailing list

4.2.3 Reply to mailing list

The Reply-To field in a mail sent to a member is the individual address of the member. Therefore, the members can reply to the mailing list in the same way as that used in the typical mailing lists. The flow when the system receives a reply mail is the same as that of a mail sent to a mailing list

4.2.4 When spam mail is received

Mailing list members judge whether the received mail is spam mail or not. Figure 6 shows the flow when a spam mail is received.

Fig. 6. Flow when spam mail is received

When a member receives a mail, he/she judges whether it is a spam mail or not. If it is a spam mail, he/she replies with a *spam report* mail. Then, by referring to the History DB, the system identifies the mail judged as spam mail from the In-Reply-To field. Next, the

individual address that has been used to send the mail is determined. Then, the penalty value of the member using the individual address is increased. Note that it is better that the penalty value is increased only when several notices are received because of a mistaken spam report.
When the penalty value of a member exceeds the threshold, a *change request* mail to change the individual address is sent to the member. The change request includes one or more candidates of the cause of the individual address leakage described in section 2.
We also can consider the use of a spam filtering tool. The filtering tool will decide whether a mail is a spam mail or not. Thus, spam mails will be blocked without sending them to the mailing list members; however, we have to pay attention to false positive and negative results.

4.2.5 Identify an address leakage caused by DHA

An address leakage caused by DHA happens without relation to members' carelessness. Therefore, we want to distinguish an address leakage caused by DHA from other cases. In DHA, an attacker will usually send spam mails to not only individual addresses but also unavailable addresses. Therefore, the system can identify DHA by monitoring the spam mails sent to unavailable addresses. Thus, dummy addresses that are similar to individual addresses are set as monitoring addresses. For example, if an individual address is xxx1@..., xxx0@... and xxx2@... are set as monitoring addresses. When a spam mail is sent not only to xxx1@... but also to xxx0@... and xxx2@..., the spam mail can be assumed to be caused by DHA. Then, even if a member receives a change request mail, he/she can know the address leakage is not caused by his/her failure. This will avoid unnecessary trouble from the change request mail.

4.2.6 Other functions

Request of a Posting History: A member can obtain his/her sent mail recorded in History DB by sending a *summary* mail to his individual address.
Withdrawal from Mailing List: When a member wants to withdraw from a mailing list, he/she sends a *bye* mail to his individual address. Then, his/her information from the Address-ML, Address-Sender, and Penalty tables is deleted, and an acceptance mail is sent to him/her.
Change of Individual Address: When a member wants to change his/her individual address, he/she sends a *change address* mail along with his/her customized address. The flow is the same as a part of the flow given in section 4.2.1.
These are optional functions, and the administrator of the mailing list can disable them.

5. Evaluations

5.1 Simulation results

We simulate the number of spam mails blocked by utilizing our system. In this simulation, we use the following configurations.

- One year is 360 days; thus, one month is 30 days.
- The individual addresses of each member have leaked out in the probability of 1/360 per day.

- Once an individual address has leaked, a spammer sends an average of one spam mail per day.
- The number of the mailing list members is **M**.
- A typical mailing list (not our mailing list system) changes its mailing list address at the end of each year.
- In our system, when each member receives a spam mail, he/she sends a spam report in the probability of **P**.
- Once our system receives **N** spam reports for a certain individual address, the individual address is changed.

Figure 7 shows the number of spam mails when the number of mailing list members is different, where **P** = 5% and **N** = 3.

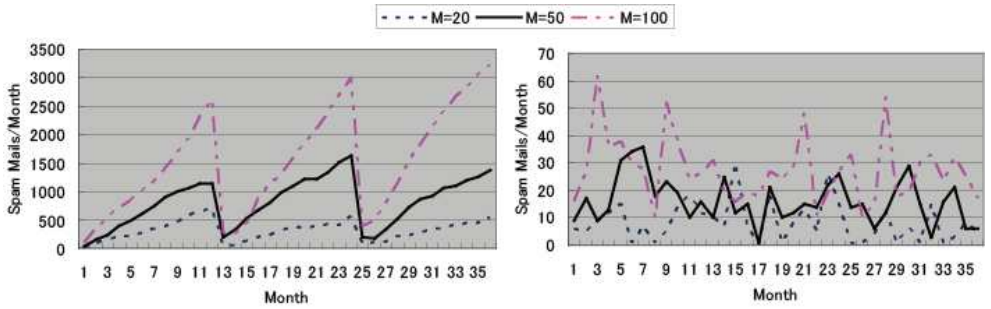

Fig. 7. Number of spam mails for different numbers of mailing list members. The left graph is the result for typical mailing list; the right graph is for our system.

In a typical mailing list, the number of spam mails increase throughout the year, and spam mails become 0 at the end of the year because of the change of mailing list address. When **M** = 20, 10.5 spam mails per day are received on an average; for **M** = 50, 27 spam mails; and for **M** = 100, 51 spam mails. However, in the case of our system does not increase the number of spam mails throughout the year because our system changes the individual addresses leaked to spammers once some spam mails (reports) are received. When **M** = 20, 0.28 spam mails per day are received on an average; for **M** = 50, 0.54 spam mails; and for **M** = 100, 0.91 spam mails. As just described, our system dramatically decreases the number of spam mails. From the viewpoint of address changes, the typical mailing list requires the change of the mailing list address every year so that the mailing list imposes the task of M person-time address changes. Our systems also requires about M person-time address changes (a little smaller than M in theory) because each member's individual address is leaked once a year on an average. Thus, the burden of each member to address changes is almost the same in both the typical and our mailing list systems. The simulation result is also almost the same as shown in table 1.

	M = 20	**M** = 50	**M** = 100
Typical mailing list	20	50	100
Proposed system	18	52.4	98.6

Table 1. Number of address changes in a year (Average of three years)

Moreover, typical mailing list systems do not identify members who leak mailing list addresses. Therefore, a careless member who leaks an address probably does not pay attention to the address leakage. On the other hand, in our system, the careless members receive a change request mail, and thus, they can pay attention to the management of their individual addresses. Thus, our system will work more effectively to reduce spam mails than the typical mailing list systems because each member will be more careful about protecting his/her individual address from leakage.
Next, we show the number of spam mails when the probability **P** is different, where **M** = 50 and **N** = 3 (figure 8).

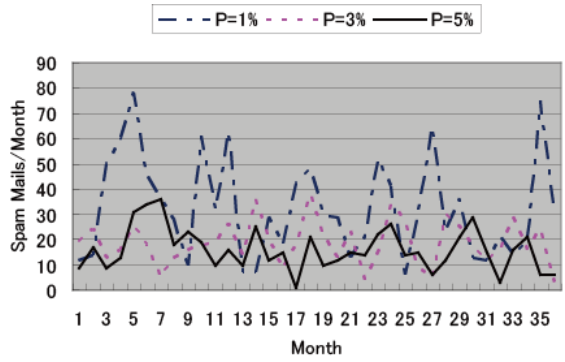

Fig. 8. Number of spam mails of our system for different probabilities of spam report

As the probability increases, the system receives spam reports with high frequency when a spam mail received. Thus, the sooner an individual address is changed, the sooner the spam mails stop. In the simulation result, even if **P** = 1% (only one person sends a spam report when hundred people receive a spam mail), 1.1 spam mails per day are received on an average. As shown here, our system can decrease the number of spam mails even when only a few members are cooperative.
The number of spam mails received when the value of **N** is different is shown in figure 9, where **M** = 50 and **P** = 5%.

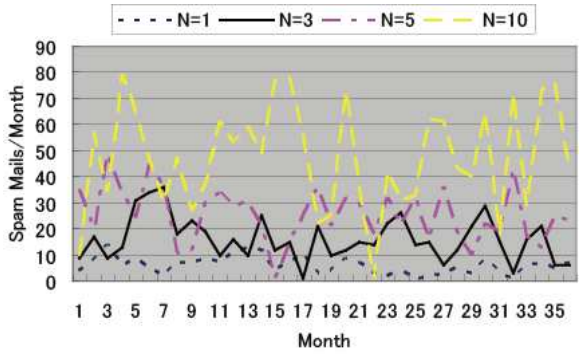

Fig. 9. Number of spam mails when **N** spam reports cause an individual address change

The lower the value of **N**, the sooner will be the individual address change. However, **N** = 1 may cause a mistaken address change, for example, even when a member mistakes a legitimate mail as a spam mail, the individual address is changed. Therefore, **N** should have an appropriate value to prevent this problem. In the simulation result, when **N** = 1; 0.20 spam mails are received per day; for **N** = 3, 0.54 spam mails; for **N** = 5, 0.85 spam mails; and for **N** = 10, 1.59 spam mails. We believe that **N** = 3 and counting spam reports for only one month would be the practical configuration.

5.2 Effectiveness in the difference in cause of individual address leakage

Figure 10 shows the causes of individual address leakage.

Fig. 10. Causes of individual address leakage

Cases 1 to 4 are caused by the carelessness of the mailing list members or the vulnerability of their machines/environments. Therefore, it may be difficult to identify the cause from these 4 cases.

Cases 1 and 2 are caused by the carelessness of the mailing list members. Therefore, the spam mails will be blocked once the individual address is invalidated since the carelessness is usually not happened frequently. In case 1, the individual address leaks from a Web page. Therefore, this cause cannot be identified by the system. In case 2, the cause can be identified if the non-member's address is written in the To and Cc fields, but not if the non-member's address is written in the Bcc field. In these two cases, the member can identify the cause by investigating his/her Web page and mail log. Then, the member can avoid such carelessness in future.

Case 3 is caused by spyware installed on the member's machine. If the spyware learns the steps involved in our system, it may execute the process of new address assignment instead of the member. Then, spam mails cannot be blocked by invalidating the individual address. Furthermore, if the spyware automatically deletes the change request mails, the member will not even notice the deletions. Therefore, if the spam mails do not stop despite changing the member's individual address several times, the administrator must inform the member that his/her address has been leaked through other means such as a telephone call. Then, the member should obtain his posting history using another machine and check it. If the member finds mails that he/she did not send/receive, it can be assumed that his/her machine is infected with a spyware.

Case 4 also cannot be solved by the invalidation of an individual address if an attacker is continuously tapping mails. In this case, the encryption of mails can solve the problem. However, members have to install software for mail encryption. This will decrease the usability of the system since members cannot easily send mails to a mailing list from Web

mails or other machines that do not have the software installed. Otherwise, we can take countermeasures similar to those in case 3.
Cases 3 and 4 are caused by the vulnerability of the member's machine or the network environment. These will cause not only spam mails but also other losses such as information leakage, deletion of important information, and stepping stone attacks. Therefore, it is advisable that the members check the vulnerability of their machines and network environments when spam mails do not stop.
On the other hand, case 5 is caused without members' carelessness. This attack is identified by using the method discussed in section 4.2.5 and does not happen frequently to same member's individual address. Therefore, spam mails will be blocked once the individual address is invalidated. Moreover, same spam mails are very like to be simultaneously posted to some members' individual addresses. Thus, the system may be able to notice the occurrence of DHA.

5.3 Usability

Our system does not require particular software to be installed on the client machines. Therefore, our system is compatible with the typical mail systems, including the protocols and mail client machines. Further, since the system does not restrict access by access controls, mailing list members can send mails to a mailing list from any mail address and any server, such as Gmail, Yahoo! Mail, and their own home mail servers. However, each member has to register his/her customized individual address. This does not seem to be much of inconvenience to the almost members because the system requires the registration only once. This would rather be useful for the members because they can use their customized individual addresses that the user can easily remember. On the other hand, this may be burdensome to a member who causes spam mails since he has to repeat this process several times. However, since he/she is the cause of spam mails, his/her discomfort seems not to be a problem. Thus, the system can stop spam mails to a mailing list but does not impose a burden on good mailing list members. Furthermore, the system is easy to use because it requires almost the same operation as the typical mailing list systems.

6. Conclusion

In this chapter, we have introduced a system to block spam mails in mailing lists, in which we assign different individual addresses to each mailing list member. When a spam mail is received, the individual address that is the cause for receiving the spam mail is identified and invalidated. Then, the spam mails from the individual address can be blocked. Furthermore, our system does not require the installation of any particular software on the client machines. Therefore, the system is highly compatible with typical mail systems. Further, mailing list members can send/reply to a mailing list by the same operation as that used in the typical mailing list systems. Evaluation results show our system is effective in reducing the number of spam mails in mailing lists.

7. Acknowledgment

This work was partially supported by the Telecommunications Advancement Foundation.

8. References

Allman, E.; Callas, J.; Delany, M.; Libbey, M.; Fenton, J. & Thomas, M. Domainkeys identified mail (DKIM) signatures, *RFC 4871*, http://www.ietf.org/rfc/rfc4871.txt, 2007.

DomainKeys Identified Mail (DKIM), http://www.dkim.org/, 2008.

DNS Blacklist (DNSBL), http://en.wikipedia.org/wiki/DNSBL, 2008.

Dwork, C. & Naor, M. Pricing via Processing or Combatting Junk Mail, *CRYPTO'92*, LNCS 740, pp. 137-147, 1993.

Fujii, M. A New Method of Spam Message Discrimination, *MS Thesis*, Waseda Univ., 2004.

Kraut, R.E.; Sunder, S.; Telang, R. & Morris, J. Pricing Electronic Mail to Solve the Problem of Spam, *Human-Computer Interaction*, Vol. 20, No. 1 & 2, pp. 195-223, 2005.

Kuipers, B.J.; Liu, A.X.; Gautam, A. & Gouda, M.G. Zmail: Zero-sum Free Market Control of Spam, *Proc. of the 25th IEEE International Conference on Distributed Computing Systems Workshop*, pp. 20-26, 2005.

Pfleeger, S.L. & Bloom, G. Canning Spam: Proposed Solutions to Unwanted Email, *IEEE Security & Privacy*, Vol. 3, No. 2, pp. 40-47, 2005.

Roman, R.; Zhou, J. & Lopez, J. Protection against Spam Using Pre-Challenges, *Proc. of 2005 IFIP International Information Security Conference*, pp. 281-293, 2005.

Seigneur, J. & Jensen, C.D. Privacy Recovery with Disposable Email Addresses, *IEEE Security & Privacy*, Vol. 1, No. 6, pp. 35-39, 2003.

Sender ID Framework (SIDF), http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx, 2008.

Symantec. The State of Spam Report, http://www.symantec.com/business/theme.jsp?themeid=state_of_spam, 2008.

Tabata, T. SPAM Mail Filtering: Commentary of Bayesian Filter, *Journal of Information Science and Technology Association*, Vol. 56, No. 10, pp. 464-468, 2006.

Takahashi, K.; Abe, T. & Kawashima, M. Stopping Junk Email by Using Conditional ID Technology: privango, *NTT Technical Review*, Vol. 3, No. 3, pp. 52-56, 2005.

Wong, M. & Schlitt, W. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1, *RFC 4408*, http://www.ietf.org/rfc/rfc4408.txt, 2006.

Multimedia

Edited by Kazuki Nishi

ISBN 978-953-7619-87-9
Hard cover, 452 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Multimedia technology will play a dominant role during the 21st century and beyond, continuously changing the world. It has been embedded in every electronic system: PC, TV, audio, mobile phone, internet application, medical electronics, traffic control, building management, financial trading, plant monitoring and other various man-machine interfaces. It improves the user satisfaction and the operational safety. It can be said that no electronic systems will be possible without multimedia technology. The aim of the book is to present the state-of-the-art research, development, and implementations of multimedia systems, technologies, and applications. All chapters represent contributions from the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821