LinuxFocus article number 279
http://linuxfocus.org

by Katja and Guido Socher
<katja/at/linuxfocus.org
guido/at/linuxfocus.org>

About the authors:

Katja is the German editor of LinuxFocus. She likes Tux, film & photography and the sea. Her homepage can be found here.

Guido is a long time Linux fan and he likes Linux because it gives you choices and freedom. You can choose and develop solutions according to your needs.

Fighting against Spam-Mail

Abstract:

Spam between your mail!? Spam E-mail is growing at an alarming rate and it is a major problem for almost everybody.
In this article we will explain what to do against this plague.

What is spam-mail?

Spam-mail has many names. Some call it UCE (Unsolicited commercial email) others call it just Unwanted E-mail but all these names don't really say what it is. If you don't get spam (yet) then take a look at this collection of spam-mail (spam_samples.html). It's a random selection of spam-mail collected over just a few days. Read through the mails and you will soon understand that it has nothing to do with commerce or business. These spammers are criminals. No serious business man/woman would annoy and offend millions of people to find a few "idiots" who would buy their tricks.

It is a common misunderstanding of people who have not much used the Internet to believe that this type of advertisement can be compared to information they get from time to time from their local supermarket. Products sold via spam-mails are often illegal or no products at all. They are tricks to get your money.

How much?

Spammers get your e-mail addresses from webpages, news groups or domain records (if you have your own domain). There are individuals who use robots to extract the addresses, burn them on CDs and sell them very cheap to other Spammers. If you write your e-mail address in clear text onto your homepage today such that programs can extract it, then you will have a major problem in a few months time and you can't stop it. The problem will be growing every day!

In 1998 the percentage of spam mail sent to LinuxFocus was less than 10%. As of November 2002 the statistics are as follows:

Our server gets about 4075 mails per week. 3273 are spam-mails!
=> **80% of all mail is Spam.**

That is 80% of the capacity of the mail server and 80% of the network bandwidth is for something that nobody wants.

Out of these 3273 spam mails about 40% originate in America (mostly Canada, US, Mexico) and about 30% in Asia (mostly Korea, China, Taiwan).

What to do with Spam

If you look at the spam-mails you will notice that almost all offer a possibility to be removed from the list. Don't do it! You are dealing with criminals. None of the spammers get anything if they maintain a proper remove list. Why do they still add this possibility? The answer is simple. It makes a much better impression on the reader and it's an excellent statistical tool. The spammers can immediately check that their mails arrive. In other words **you confirm the reception of the mail!**

There is also a simple technical problem with the idea of a remove list. LinuxFocus is not a very big site but we would need 1 person full time to unsubscribe 3273 Spam mails per week and then this person would need to unsubscribe one mail every minute . Every spammer uses a different method, it would be an idiotic task and it can't work. Remove lists are nonsense and help only the spammers.

The only right thing to do is: delete it.

Software to handle spam

There are many different options to filter out spam and this is good because it makes it harder for spammers to circumvent them. It's however an arms race. The tools to filter spam become more sophisticated but spammers improve their methods too.

There are 2 types of filters:

1. Checks directly build into the MTA (Message Transfer Agent=Mail server). Here you can usually

reject the mail. That is: you don’t even store the email. You send an error code back as soon as you recognize that this is spam during the reception of the email. Typical tools of this kind are IP based blocklists and mail header checks. If you don’t have your own Mailserver then your ISP would need to configure this.

2. Filtering after the reception of the mail. In this case the email is successfully delivered and will be filtered out later.

We will now discuss the different possibilities in detail, all of them have advantages and disadvantages. The best solution to get rid of all spam is to use several different tools.

Rejecting email directly at the MTA

If you reject your mail directly at the mail server during the reception of the mail then the spammer can get back an error code and knows that this address does not work. If he is one of the "CD-makers" then he might take out the address. It can save network bandwidth because you don’t have to receive the full message. You can send the error code back as soon as you find that this is spam.

To do this you need a good and flexible MTA. Unfortunately the two most common servers, Sendmail and the one from Bill Gates are not good at all for this task. Two very good alternatives are Postfix and Exim. If you can’t change your server then you can put an smtp proxy such as messagewall in front of the server (smtp = Simple Mail Transfer Protocol, the Internet mail protocol).

We will now discuss some common filter techniques and how they work. We will not describe how to configure them exactly in each MTA. It would make the article too long. Instead we suggest to read the documentation that comes with the MTA that you have installed. Postfix and Exim are well documented.

- Realtime Block lists:
  These are DNS based lists. You check the IP address of the mailserver that wants to send mail to your server against a blacklist of known spammers. Common lists are www.spamhaus.org or ordb.org. There is also a tool called blq (see references) to manually query such block lists and test if a given IP address is listed. You should however not be too enthusiastic about it and carefully choose the lists since there are also some which block entire IP ranges simply because one spammer had used a dialup connection from this ISP at one point in time. We personally would at least enable ordb.org to keep out mail from poorly administrated servers.
  Experience shows that these lists block about 1%-3% of the spam mail.

- 8 bit characters in subject line:
  About 30% of the spam origins in China, Taiwan or other Asian countries these days. If you are sure that you can’t read Chinese then you can reject mail which has a lot of 8 bit characters (not ASCII) in the subject. Some MTAs have a separate configuration option for this but you can also use regular expression matching on the header:

  ```
  /^Subject:.*[^ -~][^ -~][^ -~][^ -~]/
  ```

  This will reject email which has more than 4 consecutive characters in the subject line which are not in the ASCII range space to tilde. If you are not familiar with regular expressions then learn

them, you will need them (See LinuxFocus article 53). Both exim and postfix can be compiled with perl regular expression support (see www.pcre.org). Perl has the most powerful regular expressions.
This method is quite good and keeps out 20-30% of the spam-mail.

- Lists with "From" addresses of known spammers:
  Forget it. This used to work back in 1997. Spammers today use faked addresses or addresses of innocent people.

- Reject non FQDN (Fully Qualified Domain Name) sender and unknown sender domain:
  Some spammers use non existent addresses in the "From". It is not possible to check the complete address but you can check the hostname/domain part of it by querying a DNS server.
  This keeps out about 10-15% of the spam and you don’t want these mails anyhow because you would not be able to reply to them even if they were not spam.

- IP address has no PTR record in the DNS:
  This checks that the IP address from where you get the mail can be reverse resolved into a domain name. This is a very powerful option and keeps out a lot of mail. We would not recommend it! This does not test if the system administrator of the mail server is good but if he has a good backbone provider. ISPs buy IP addresses from their backbone providers and they buy from bigger backbone providers. All involved backbone providers and ISPs have to configure their DNS correctly to make the whole chain work. If somebody in between makes a mistake or does not want to configure it then it does not work. It says nothing about the individual mail server at the end of the chain.

- Require HELO command:
  When 2 MTAs (mail servers) talk to each other (via smtp) then they first say who they are (e.g. mail.linuxfocus.org). Some spam software does not do that. This keeps out 1-5% of the spam.

- Require HELO command and reject unknown servers:
  You take the name that you get in the HELO command and then you go to DNS and check if this is a correctly registered server. This is very good because a spammer who uses just a temporary dialup connection will usually not configure a valid DNS record for it.
  This blocks about 70-80% of all spam but rejects also legitimate mail which comes from sites with multiple mail servers where a sloppy system administrator forgot to put the hostnames of all servers into DNS.

Some MTAs have even more options but the above are quite commonly available in a good MTA. The advantage of all those checks is that they are not CPU intensive. You will usually not need to update your mailserver hardware if you use those checks.

Filtering of already received mail

The following techniques are usually applied to the complete mail and the mail server who sends the mail does not notice that the mail could not be delivered. It means also that a legitimate sender will not get a failure report. The message will just disappear.
Having said this we must also say that this is not totally correct because it really depends on the filtering possibilities of the mail server. Exim is very flexible and would allow you to write custom filters on

messages.

- SpamAssassin (http://spamassassin.org/):
  This is a spam filter written in perl. It uses carefully handwritten rules and assigns certain points to typical spam phrases such as "strong buy", "you receive this mail because", "Viagra", "limited time offer".... If the points are above a given level then the mail is declared as spam. The problem with this filter is that it is very heavy in terms of memory and cpu power. You will probably need to upgrade your mail server hardware especially if the server is already 2-3 years old. We would not recommend to use it directly on the mail server. Spamassassin comes with a spamd program (spamd=spam daemon + spamc=client to connect to the daemon) which will reduce the startup time of spamassassin and reduce the cpu consumption but it is still a very resource demanding application.

  To filter the mail you need to create a .procmailrc file (and .forward) similar to this one:

  ```
  # The condition line ensures that only messages smaller than 50 kB
  # (50 * 1024 = 56000 bytes) are processed by SpamAssassin. Most spam
  # isn’t bigger than a few k and working with big messages can bring
  # SpamAssassin to its knees. If you want to run SpamAssassin without
  # the spamc/spamd programs then replace spamc by spamassassin.
  :0fW:
  * < 56000
  | /usr/bin/spamc
  # All mail tagged as spam (e.g. with a score higher than the set threshold)
  # is moved to the file "spam-mail" (replace with /dev/null to discard all
  # spam mail).
  :0:
  * ^X-Spam-Status: Yes
  spam-mail
  ```

  The installation is easy and spamassassin will filter more than 90% of the spam.

- procmail (http://www.procmail.org):
  Procmail is not a spam filter on its own but you can use it to write yourself one. procmail is also very light weight as long as you limit the number of rules to something reasonable (e.g. less than 10). To use it you create a .forward file in your home-directory and add there the following line:

  ```
  "| exec /usr/bin/procmail"
  ```

  Some people recommend to use
  "|IFS=’ ’ && exec /usr/bin/procmail"
  but this creates new problems with an extra process being created which does not run under the control of the mailserver any longer. Secure mail servers like postfix or exim will have no problems with the .forward file as shown above.

  Procmail is especially useful in an environment where you normally communicate just in a closed group. E.g. for people in a company where most of the mail should come from your colleagues and some known friends. Here is an example for "mycompany.com":

  ```
  # .procmailrc file.
  # search on header for friends:
  ```

```
:0 H:
* ^From.*(joe|paul|dina)
/var/spool/mail/guido

# search on header for mails which are not coming from
# inside mycompany.com and save them to maybespam
:0 H:
* !^From.*(@[^\@]*mycompany\.com)
/home/guido/maybespam

# explicit default rule
:0:
/var/spool/mail/guido
```

This makes it much easier to delete spam and you don't find the ugly spam between your normal mail.

Procmail is very flexible and can also be used for other tasks. Here is a totally different example: Procmail comes with a "reply to sender" program called formail. This can e.g. be used to send a message back to people. A major plague are those e-mails with word documents inside. If you are a Linux developer using e-mail to exchange information about your projects or Linux in general then you are for sure not interested in people who write text into a word document and attach it to mails. Viruses can easily be spread that way. They don't usually infect Linux but it's a bad idea in general to use MS-word for sending text to other people as it requires MS word with the same version on the receiver side to read the text. There are open formats such as RTF or HTML which do not spread viruses, are cross platform, and do not have such a version problem.

```
# Promail script to
# reject word documents. Reject the mail, but do not reply to
# error messages "From MAILER-DAEMON"
# If you use ":0 Bc" instead of ":0 B" then you will still get the mail
:0 H
* !^From.*DAEMON
{
  # The mime messages with word documents look like this in the body
  # of the message:
  #------=_NextPart_000_000C_01C291BE.83569AE0
  #Content-Type: application/msword;
  #        name="some file.doc"
  #Content-Transfer-Encoding: base64
  #Content-Disposition: attachment;
  #        filename="real file.doc"
  :0 B
  * ^Content-Type:.*msword
  | (formail -r ; cat /home/guido/reject-text-msword ) | $SENDMAIL -t
}

# explicit default rule
:0:
/var/spool/mail/guido
```

The text file /home/guido/reject-text-msword should contain a text explaining that msword documents can spread viruses and ask the sender to send the document e.g. in RTF format.

How to use procmail and what all these strange letters in the configuration file mean is very well explained in the "procmailrc" man page.

- bogofilter (http://www.tuxedo.org/~esr/bogofilter):
  Bogofilter is a Bayesian spam filter. It is entirely written in C and it is very fast (compared to SpamAssassin). A Bayesian filter is a statistical filter that you have to train first to learn what is spam and what is not spam. You need about 100 training messages (sorted into spam and not spam) until the filter can work efficiently on new messages.

  Bogofilter is fast but it does not work from day one as SpamAssassin. After a while it will be as efficient as SpamAssassin and filter more than 90% of the spam.

- razor (http://razor.sf.net/):
  This is a distributed, collaborative, spam detection system. Checksums of known spam messages are stored in a database. If you get a new mail you compute the checksum and compare it with checksums in the central database. If the checksum matches then you can discard the message as spam. razor works because special e-mail accounts have been spread over the Internet only for the purpose of getting into the address lists of all the spammers. These accounts catch only spam and no normal mail. In addition people can of course send mails to razor for marking it as spam. There is a good chance that the mails are already known as spam before they arrive in your mailbox. The system filters about 80% of the spam. razor has one characteristic that none of the other post processing and filtering techniques has: razor detects almost no false positives. That is: the number of mails which are not spam but still declared spam is very low with razor.

There are many more possible solutions to fight against spam. We believe that the above covers the most important ones.

The best solution is to use checks in the MTA as a first stage and then kill the remaining spam in a second stage with a post-processing filter.

HTML mails

A particularly dangerous form of e-mail are spam mails in HTML format.

Most spammers use the "unsubscribe possibility" to see how many of their mails arrive. HTML formatted mail offers a much better form of feedback: Images. You can compare this system with the visitor counters as found on some webpages. The spammer can exactly see when and how many of the mails are read. If you study Spam carefully you will see that in some cases the URL for included images contains a sequence number: The spammer can see exactly who looks at the mail and at what time time. An incredible security hole.

Modern mail reader programs will not display images which are downloaded somewhere from a URL. However there is hardly any modern and secure HTML mail reader. Kmail and the very latest version of mozilla mail offer the possibility to disable images from external sources. Most other programs will generate nice statistics for the spammer.

The solution? Don’t use a html mail capable program or download the mail first then disconnect from the Internet and then read the mail.

Where does the spam come from?

Never trust the sender address in the "From" field of spam mails! These are either non existent users or innocent people. It is very rare that this is the mail address of the spammer. If you want to know where the mail comes from then you have to look at the full header:

```
...

Received: from msn.com (dsl-200-67-219-28.prodigy.net.mx [200.67.219.28])
        by mailserver.of.your.isp (8.12.1) with SMTP id gB2BYuYs006793;
        Mon, 2 Dec 2002 12:35:06 +0100 (MET)
Received: from unknown (HELO rly-xl05.dohuya.com) (120.210.149.87)
        by symail.kustanai.co.kr with QMQP; Mon, 02 Dec 2002 04:34:43
```

Here an unknown host with IP address 120.210.149.87 who claims to be rly-xl05.dohuya.com sends the mail to symail.kustanai.co.kr. symail.kustanai.co.kr sends this message further on.
The spammer is hiding somewhere behind 120.210.149.87 which is probably just a dynamic dialup IP address.

In other words the police could find this person if they would go to the owner of kustanai.co.kr and ask for server logs and a printout of connections from the local telephone company. You have very little chance of finding out who that was.

It could also be that the first part is faked and the spammer is really behind dsl-200-67-219-28.prodigy.net.mx. This is very likely since there is no good reason why symail.kustanai.co.kr should send the mail to msn.com via the dsl dialup connection (dsl-200-67-219-28.prodigy.net.mx). The mailserver.of.your.isp (symbolic name) is the server of your Internet Service Provider and is the only part from this "Received:" line which is reliable.

It is possible to find the spammer but you need international intelligence and police forces to go to prodigy.net.mx.

Conclusion

If spam continues to increase at the current rate then the Internet will soon transport a lot more Spam than real e-mail. Spam is transported at the cost of the receiver. More bandwidth is needed and often the mail systems need to be upgraded to handle the Spam.
Laws in many countries do little to protect people against criminal spammers. In fact some countries have laws which restrict only honest people (digital rights management etc. ...) and help the criminals (e.g. to get nice statistics about the spam-mail).

Join the Coalition Against UCE!

EuroCAUCE
Fighting European Spam

http://www.euro.cauce.org/en/

CAUCE
Coalition Against Unsolicited Commercial Email
Take back your mailbox

http://www.cauce.org/

Internet Service Providers should check their mail systems. No unauthenticated access to mail servers must be given and the amount of mails that one user can send per minute must be limited.

References

- http://spamassassin.org/: spamassassin homepage
- http://www.procmail.org/: procmail homepage
- http://www.spambouncer.org/: spambouncer: a procmail based spam filter
- http://www.postfix.org/: homepage of the postfix MTA
- http://www.exim.org/: homepage of the exim MTA
- http://messagewall.org/: homepage of the messagewall smtp proxy
- http://www.unicom.com/sw/blq/: the blq perl script to query DNS based block lists
- http://www.ordb.org/: DNS based open relay block list
- http://www.spamhaus.org: DNS based block list
- http://www.samspade.org/: Where does the spam come from?
- http://www.dnsstuff.com/: various blocklists and DNS based tools
- http://www.geektools.com/cgi-bin/proxy.cgi: geektools Whois proxy
- http://www.tuxedo.org/~esr/bogofilter/:bogofilter mail filter
- http://razor.sf.net/: razor
- http://pyzor.sourceforge.net: razor implemented in python
- http://lwn.net/Articles/9460/: Linux weekly news article comparing bogofilter and spamassassin.

Webpages maintained by the LinuxFocus Editor team © Katja and Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: en --> -- : Katja and Guido Socher <katja/at/linuxfocus.org guido/at/linuxfocus.org>

2005-01-14, generated by lfparser_pdf version 2.51