ForLinux
Open Source Support, Hosting and Training

Table of Contents

ForLinux
Open Source Support, Hosting and Training

Introduction

This paper will discuss the best practices for stopping the maximum amount of SPAM arriving in a user's inbox.

It will outline simple but effective measures for hardening your servers, along with some countermeasures. This paper is aimed at system administrators who manage Linux servers and have an understanding of Linux command line.

A link to the home page, or a relevant web site, for each major topic and application discussed will be found in the appendix at the end of this white paper.

ForLinux
Open Source Support, Hosting and Training

Overview

Most people have some form of SPAM checking, either on the server side, or on the local clients machines.

It is best method to block SPAM as early as possible in the delivery process. For example, rather than making your clients block it on their local machine, you should simply block it when the SPAMMER first tries to connect to your mail server.

Its takes server time and power to process emails, so the earlier in the delivery process that the SPAM email can be found and deleted the better. You can save CPU cycles and bandwidth by dropping the connection before the actual email content has arrived.

Definitions :

SPAM - Unsolicited bulk email. Email designed to try sell you a product or direct you to a fraudulent website. The vast majority of email is now considered to be SPAM.

HAM - Legitimate or NON-spam emails.

Virus - Malicious code that masquerades as a legitimate image or program. Designed to do damage to a persons Computer (Operating System).

Phishing - These are emails that appear to come from a legitimate banking (or other) website and requesting you to change your password or update your credit card details. The email will contain a link to the fraudsters website which will appear genuine. Your details will be harvested and used for fraudulent activity.

False Positive - When a non-spam email is classified as SAPM, this is called a false positive. There is a constant battle to keep false positives to an absolute minimum.

With the increase in bank fraud due to PHISHING email, SPAM is not only a nuisance it can also lead to financial fraud !

Whenever a machine (or program) is tasked to try work out which emails are SPAM and which are legitimate, things can go wrong. If the machine thinks a legitimate email is SPAM, this is called a FALSE POSITIVE. There is a constant balancing act between blocking the maximum amount of SPAM and getting the lowest possible FALSE POSITIVE rate.

We are going to be reviewing a few techniques in order to prevent as much SPAM as possible reaching its target.

Common techniques to identify SPAM

Over the years spammers have been constantly evolving new methods to get their emails through. It started with simple text messages, then they started using weird spellings in order to bypass the spam traps. The newest form is sending all the information as an image, as most spam trappers can't decipher the text on the images. Below is am example of an IMAGE-based spam email.

The most common server-side method to block spam is by using a program like **Spamassassin,** which uses rules to detect if an email is spam or not. An email is processed and matched against the rules. Each rule will assign a SCORE to the email, if the overall score is higher than a pre-set limit, the email is marked as spam and the appropriate action is taken.

Just by installing Spamassasin, and using its set of default rules, quite a large portion of spam can be blocked. But there are other methods we can use before the email is accepted by our system.

Blocking SPAMMERS at connection time :

RBL's (Realtime Blackhole List's)

RBL's or Realtime Blackhole List's are precompiled lists of KNOWN SPAMMERS that you can use on your server to block the spammers.

They way it works is when a new connection is made, your email program checks the connecting IP against the RBL list, if the IP is in the list it is immediately REJECTED. This is quite an effective method as the spam email is DROPPED immediately after a connection is made. But this will also block legitimate emails coming from a blacklisted source.

Greylisting

Greylisting is a method of temporarily rejecting any email from a sender it does not recognize. Normal mail server will simple hold onto the email and then try connect again later. If a sufficient amount of time has elapsed, your server will accept the new email. Usually the Spammers have to send out to thousands of addresses, that they won't try to resend the email.

This method will delay most emails arriving (even legitimate ones) so its not the best solution to implement.

Dictionary Attack

A Dictionary Attack is when a spammer opens a connection to your server and then tries to send emails to a few different email addresses. These addresses will be to random email addresses or will come from a home made dictionary the spammer has.

An example of a dictionary attack would appear in the mail logs as :

jack@domain.com

jill@domain.com

john@domain.com

john.smith@domain.com

Catchalls

Catchalls are the single biggest cause of spam and spam delivery on a server.

Normally you set up individual email addresses for each user. An email addressed to a non-exiting email address will be rejected. This is the best method to use.

If a catchall is setup, all emails to non-existent email addresses will be accepted by the server and then have to be scanned and delivered to a mailbox. So you are basically giving the spammers a nice easy target as they DON'T need to use valid email address to get their spam through. The is forced to accept each email in for processing, which will cause the load on the machine to rise as it it churns through a huge amount of spam !!

The above methods have all dealt with blocking the SPAMMER before the contents of the email has been received.

Now we will look at the actual scanning and marking of the emails.

From address

The ”From:” address of an email can easily be spoofed, so always treat it with a certain

amount of scepticism. Often the “From:” address will even be set to the same as the “To:” address in hopes it will get through easier.

HELO / EHLO

You can also configure your email program to reject any mail servers that connect to it and then give an invalid HELO or EHLO. This is a basic greeting all email servers should give them connecting. Often the spammers will use poorly configured mail servers, so this will prevent spam at server connection time.

SPF records

SPF or Sender Policy Framework uses DNS entries to list the actual server that can send email for a domain. For example “**forlinux.co.uk**” will publish the IP addresses of our mail servers. Any email an end user receives that has come from an IP can matched up with the ones we publish. If the IP does NOT match, it means the email did not come from ForLinux.

Detecting SPAM

The most common program on Linux to block SPAM is called **Spamassassin**. It is responsible for accepting an email onto the server, scanning it and then deleting or delivering it depending on the rules matched.

Each email scanned is matched against a ruleset and assigned a score. This score will determine what Spamassassin does with it. There are three basic categories of email - **Non-spam**, **Normal spam** and **High Scoring spam**.

As expected Non-spam is an email that Spamassassin thinks is clean and can be delivered to a users mailbox.

High Scoring spam is an email that has matched quite a few rules an is almost 100% guaranteed to be SPAM. These are usually just deleted.

Normal Scoring spam is the email that sites in the middle between Non-spam and High Scoring. These are the emails that could be SPAM, but might also be legitimate. These are best not simply deleted, but instead delivered to the user, but the emails **Subject** line will be modified to begin with the words “**{SPAM?}**”.

The end user can then use this to send them directly to a spam or other mailbox and only check them occasionally.

Each of the Spamassassin rules that are triggered are added up and the total determines how an email is categorised. The higher the score, the more chance of it being spam.

A good starting point is :

•Anything below a score of 5 is Non-Spam.

•Between 5 and 15 is Normal Spam and,

•Over 15 is High Scoring spam.

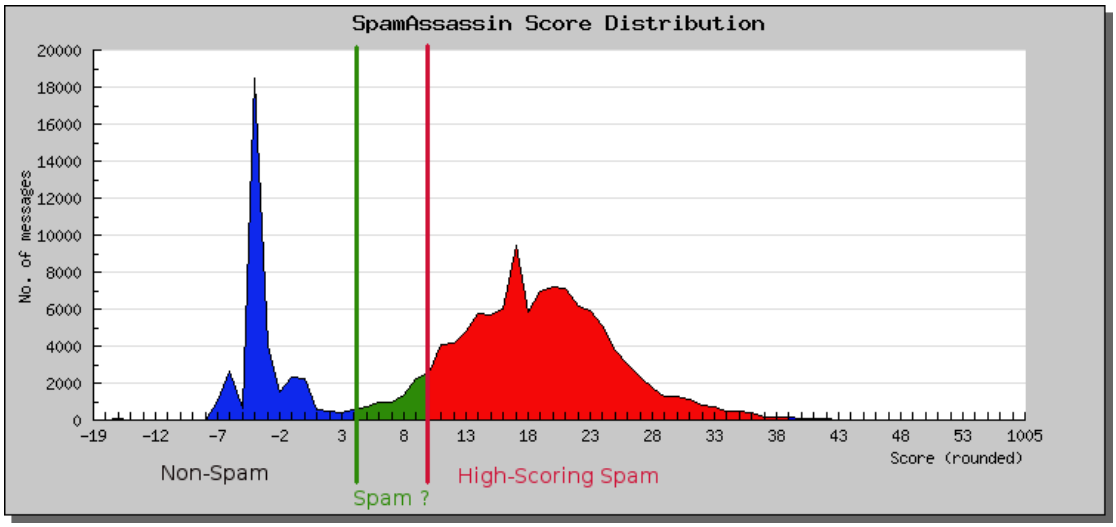
Below is an image showing the distribution of SPAM by score. (Lower is better).

•The BLUE shaded graph is legitimate email (about 35%).

•The GREEN shaded graph are possible spam or legitimate emails (about 5%).

•The RED shaded graph is regarded as High-Scoring spam and is usually just deleted on detection (about 60%).

The diagram shows the huge portion of email received that is actually SPAM !!



ForLinux
Open Source Support, Hosting and Training

MailScanner www.mailscanner.info

Spamassassin is best installed with **MailScanner**.

MailScanner proves an easy to use configuration file where all the SPAM scores and rules can be configured. MailScanner will be responsible for then passing the emails to Spamassassin for scanning.

MailScanner also has support for other Plugins :

•**Anti-virus** - the most common free one in Linux is **Clam Antivirus**. This will ensure emails with viruses are blocked as-well. Many other virus scanners are support (see the MailScanner.conf file). The latest virus patterns are also update on a daily or more regularly basis. Most viruses cannot be safely removed, so they are usually just deleted.

•**Phishing emails** - MailScanner will check all URLs in the email against a pre-set list of known phishing sites. It will then mark the subject line with **{Fraud?}**.

Each of the above can be configured as to whether it should deliver the email, quarantine it or delete it. A copy can also be sent to a third party email address.

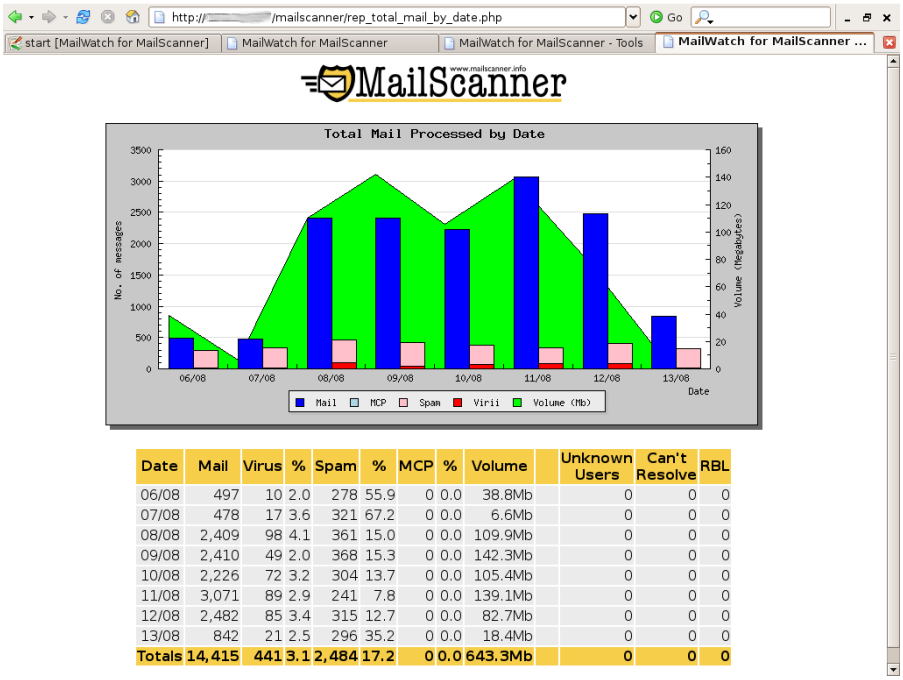
How is spam identified ?

Bayesian filters are used, they are based on algorithms that use probability to block unwanted messages. the algorithm will look through the email and then using rules try predict whether its spam or not. For example, spam emails will often contain the word “Viagra”.

Applied to emails, this means that if you break a message down into its basic elements (text, HTML, URLs) and you find that particular elements recur frequently in spam emails and not in normal mail. This indicates a degree of confidence that the message indeed spam. The email can then be scored against the matches.

MailWatch is a web-based front-end PHP. It shows the latest emails scanned, which have been marked as spam etc.

It also produces summaries of SPAM and viruses.



Date	Mail	Virus	%	Spam	%	MCP	%	Volume		Unknown Users	Can't Resolve	RBL
06/08	497	10	2.0	278	55.9	0	0.0	38.8Mb		0	0	0
07/08	478	17	3.6	321	67.2	0	0.0	6.6Mb		0	0	0
08/08	2,409	98	4.1	361	15.0	0	0.0	109.9Mb		0	0	0
09/08	2,410	49	2.0	368	15.3	0	0.0	142.3Mb		0	0	0
10/08	2,226	72	3.2	304	13.7	0	0.0	105.4Mb		0	0	0
11/08	3,071	89	2.9	241	7.8	0	0.0	139.1Mb		0	0	0
12/08	2,482	85	3.4	315	12.7	0	0.0	82.7Mb		0	0	0
13/08	842	21	2.5	296	35.2	0	0.0	18.4Mb		0	0	0
Totals	**14,415**	**441**	**3.1**	**2,484**	**17.2**	**0**	**0.0**	**643.3Mb**		**0**	**0**	**0**

Prevention

Use multiple email addresses :

•For normal clients give them your proper email address.

•For websites and postings online - use another email address.

This will allow you to protect your real email address from spam. You can also set up temporary email addresses that you know will quickly get spammed, and then simply remove them later. The server can use more stringent checks on your “temporary” email addresses.

Summary

Prevention is still the best solution here, but the detection steps discussed above help block any spam that does get through. Education users is anther key step. Make sure they don't blindly just click on every email and link they receive. Only email from trusted people should be responded to.

Just by following the simple rules laid out above, you should be able to keep your precious data safe and secure.

Appendix

Spamassassin http://spamassassin.apache.org/

ClamAV - http://www.clamav.net/

MailScanner - http://www.mailscanner.info/

MailWatch - http://mailwatch.sourceforge.net/

Sender Policy Framework - http://www.openspf.org/