Fighting spam in a corporate environment using open-source solutions – a case study

Pieter Blaauw (pblaauw@pnp.co.za)
Pick 'n Pay Information Systems

Abstract

The problem of spam is one that many organizations face today, and there are virtually no e-mail users that have not suffered from receiving unsolicited e-mail. Simply put, spam is symptomatic of an inefficient and parasitic business; one that cannot bear the cost of its own activities, and thus moves the cost of its activities on to others. More importantly, the cost of filtering spam is totally disproportionate to the benefit of the spammer. E-mail users at Pick 'n Pay suffered from this same problem, with some cases being as high as half of all e-mail being spam. This paper discusses the path followed by Pick 'n Pay Information Systems in finding a solution to this ever growing problem.

Pick 'n Pay looked at several solutions in order to find the most cost effective way of filtering spam for the users. The most cost effective solution was the implementation of an open-source spam filter, based on a high availability solution designed for future scalability. The solution provided us with the ability to filter e-mail at the ISP, thus saving expensive bandwidth, yet allowing users to administrate their own quarantine boxes and providing them with the ability to update the software to create better accuracy with regards to spam filtering. The solution proved that open-source solutions can work in a corporate environment with the proper testing and implementation. Current statistics provide us with a filtering rate accuracy of 98% with less than 0.002% false positive rate.

Key Words
spam, fighting spam, open source, DSPAM

1. Introduction

There are very few things modern computer users find more annoying than spam. Whether it is in the middle of a busy workday, or when you come in to the office first thing in the morning after a weekend, chances are there will be spam in your mail. These messages range from get rich quick schemes, selling of imitation goods, to advertisements of rather questionable pharmaceutical products. Junk e-mail or spam is a growing problem for Internet users, be it individuals or large corporations.

■ What is spam?

Spam is junk e-mail that is sent to you by someone who has no prior or existing relationship with you. No matter what you call it (junk mail, unsolicited mail), spam is defined by the fact that recipients did not solicit the mail or divulge their e-mail addresses for the purpose of receiving such mail (Falk J.D. - 1998). Unfortunately each day, thousands of spam programs scan web pages, newsgroups and other online documents to harvest e-mail addresses for the purpose of sending these e-mails.

■ Fighting back!

It's pretty obvious that spammers are spending a great deal of time and effort in finding ways to use other people's resources to send junk mail and conceal themselves. Given this, what can corporates do to fight back? The following paper is a case study regarding the strategy that Pick 'n Pay Information Systems used to fight back!

2. Background

Like any other big corporate, Pick 'n Pay suffered from large amounts of spam. The only defense was the use of Trend Micro's Interscan Message Security Suite (Trend Micro – 2004). While the product is sold as a combined anti-spam, anti-virus and content filtering solution, it strength lies in its anti-virus and content filtering capability, and not in fighting spam. Trend Micro's IMSS uses only a heuristic spam identifying engine to try and catch spam coming through. The biggest problem with the implementation was that it was based on the local end of the company's Internet link. Once the spam reached the local network it was only stopped, wasting valuable bandwidth.

Another problem was dealing with stopped (quarantined) mail. It's the task of a single person to go through the large amounts of quarantined mails to find any false positives and release those. These included e-mails blocked by the content filter as well as spam. If a user suspected that a valid business mail has been blocked (false positive), they needed to log a call with the help desk to find and release that mail. This became a full time job for one person considering the amount of e-mail processed every day.

What was needed was a pure spam filtering solution on the remote end of Pick 'n Pay's Internet link, with the ability to learn from 'false positives' as well as learn from 'spam misses'. The ideal solution would also allow each user to check their own quarantined mail without having to burden the help desk with additional calls.

3. Finding and implementing a solution

Pick 'n Pay was already paying for the rent of a cabinet at UUnet with a remote router at that end. This was done to allow services like Quality-Of-Service over the link for better bandwidth management. Installing a spam filtering solution at the remote end of the Internet connection was the logical way to go about trying to stop spam from reaching the company and wasting valuable bandwidth.

3.1 Finding the right product / solution

Finding a solution to a problem as complicated as spam does not include just one individual in a company, but a whole team of staff. In the case of Pick 'n Pay Information Systems, the project team involved staff from:

- Systems Administration
- Network Engineering
- Research and development
- Information Security
- Project Administration

The Security Administrator was overall in charge of the project and the primary contact for the project office. A project was registered with the project team and the required documentation was done. This allows for a clear project definition, project scope as well as time lines and proper budget control. Proper project scope was required to prevent project goals from creeping. A good example

would be requests for the solution to be able to do virus scanning and thus deviating from the original project definition and scope.

Finding the right solution required that we define a few criteria first before looking at commercial or open source solutions to stop spam. Some of the criteria / requirements we looked at were:

- Linux / Unix based?
- User managed quarantine boxes
- Security
- User friendliness / ease of use
- Can the software be trained?

Unfortunately most of the commercial solutions on the market runs either in a 'black box' configuration (and is rather expensive), or runs on Microsoft Windows. Considering that the solution would have to live in a fairly insecure environment facing the Internet it was the general consensus that a Unix-based solution would be a better option.

3.2 Why Open-Source?

Pick 'n Pay has always been a supporter of open-source solutions, and this was an ideal opportunity to look at what was available for fighting spam and proving that open-source is indeed viable in a commercial arena. With the intent to install the servers at a remote location, security was of great concern, and the project team decided with the quick security patch cycle of open-source software that it would be the better solution. In the event that a security flaw in any of the components were detected, a open-source solution would allow us to patch and close the security vulnerability far quicker than a commercial solution (Glass B. - 2000).

Looking on the Internet, and speaking to several parties in the know from various open-source newsgroups, a few recommendations were made. These were 'Spamassasin', 'POPFile', and 'DSPAM'. Previous experiences with Spamassasin proved that it didn't allow for easy manipulation of quarantined mail. POPFile again acts like a proxy server, sitting between your mail client and your mail server, so it wouldn't make for a practical implementation in our environment (being the remote side of the Internet connection). This does not mean that any of these products aren't good. They are very good, but they did not meet the requirements set out by the project team. A closer look at DSPAM found that it might do what was required, if implemented correctly. DSPAM uses a variety of different ways to identify spam, including Chained Tokens (Zdziarski J.A. - 2004), Neural Networking, Bayesian Noise Reduction (Zdziarski J.A. - 2004), and External Inoculation (Zdziarski

J.A. - 2004).

- Chained Tokens - Chained Tokens (also known as multi-word tokens and n-grams) is a simple data processing algorithm designed to provide more specific (and much better) data for the existing statistical algorithms to work with. As the name implies, this algorithm is based on the concept of chaining adjacent tokens together to make new tokens. This approach creates 2n-2 (ish) additional data points to work with.
- Neural Networking - A type of artificial intelligence that attempts to imitate the way a human brain works. Rather than using a digital model, in which all computations manipulate zeros and ones, a neural network works by creating connections between processing elements, the computer equivalent of neurons. The organization and weights of the connections determine the output.
- Bayesian Noise Reduction - Bayesian Noise Reduction is a statistical approach to evaluating coherence by instantiating a series of machine-generated contexts to serve as a means of contrast. This makes it possible to identify text that is out of context using a form of pattern consistency checking. BNR attempts to solve the problem commonly referred to as "Bayesian Noise" which, in its simplest definition, refers to irrelevant data present in a message being classified. Bayesian Noise Reduction dubs irrelevant text in order to provide cleaner classification and is implemented as a pre-filter to existing language classification functions.
- External Inoculation - The theory behind external inoculation is this: why put _anyone_ through the misery of being the first to receive a new spam when you can have the spammers themselves send it directly to you. On top of this, external inoculation can be combined with internal inoculation by taking the spam you received externally and inoculating your friends with it internally. External inoculation is accomplished by creating a covert, external alias that is configured to automatically inoculate your dictionary from any messages it receives. The covert alias can then be published onto a series of public newsgroups and websites where it is sure to be harvested by a spammer's tools.

3.3 Testing

The next step was to gather a trial user-base and test it in a production environment. A general invitation was sent to all Information Systems staff, as well as some high level board members that were suffering from large amounts of spam to participate as test users. The risks were clearly spelled out, and about fifteen staff members volunteered. A simple low-end 'white box' pc was used as a test server running on Red Hat Linux.

Mail was simply routed through the DSPAM test-server before being passed on to the Microsoft Exchange server. The difficulty we experienced in finding out how effective the filtering was being done was due to Trend IMSS running in the DMZ already. The fifteen test users were then put in a separate group on IMSS and mail was not filtered for spam, but purely for viruses and content. This allowed us to fairly accurately gage how much spam was getting through and how much spam was being caught by DSPAM.

The trial period was run for a period of fourty-five days. This was enough time for DSPAM to learn from the spam passed to it, and for the users to see how effective it was. Within two weeks many of the test persons reported a significant drop in the amount of spam they received. A few more users requested permission to be added to the test server and again, within days they reported a marked drop in spam reaching them.

During the test period, only one problem occurred. DSPAM had a bug where it would malformat certain messages, but this was soon solved and the new version was installed. After the trial period a questionnaire was sent to the test users, and the feedback was very positive.

Users liked the idea of having their own 'quarantine' mailbox with spam, and being in control of when and how fast a false positive (very few occurred) could be released. Users were also able to control how strict the spam filter was applied, ie, very strong (less spam, more false positives), or even very weak (more spam, few false positives). It must be said that the false positive rate was extremely low during the test period. With a successful test, it was decided to look in to putting the solution in to production.

3.4 Perform and Predict

Unlike the usual approach of 'buy the biggest server possible', a proper capacity plan for the server (s) were needed and the hardware acquired according to the data received. For this task BMC Software was used, in particular BMC Patrol. BMC is used extensively within Pick 'n Pay to monitor server status and to do capability planning.

"PATROL for Unix - Perform & Predict helps you understand the past, present and future performance of your IT environment. Historical analysis provides the proper business perspective for IT decisions. Real-time analysis ensures your ability to identify and resolve current application bottlenecks and performance issues. Analysis and predictive modelling identifies trends, providing

rapid recognition of normal and abnormal usage, and reveals the impact of changes on your business. " (Taken from the PATROL® for Unix - Perform & Predict brochure supplied by BMC.)

The BMC agent was installed on the test server and in turn sent raw data back to a central server from where the Enterprise Resource Team could do a 'perform and predict'. This process looks at the data gathered for a certain amount of users and e-mail processed by the test server and then calculates what the requirements would be for a larger number of users and e-mails. The agent was set to collect data for a fourteen day period, after which a 'perform and predict' report was done. The recommendations in the report were followed as far as the purchasing of a scalable platform for the DSPAM servers. The report indicated that due to high I/O, very fast disk storage was the greatest requirement in the specified hardware.

Due to the reliance on e-mail, and the fact that the solution would be installed off-site it was decided by the project team that it would be better to run a high-availability solution consisting of two servers. This would allow for complete fail-over, since each server had the ability to easily mirror drives, and were also purchased with dual power-supplies.

3.5 Installing and testing

The operating system used was Red Hat Enterprise Linux AS, running Postfix, Mysql, Apache and DSPAM. Once each server was installed and the high-availability was properly tested, the servers were moved and installed at the data-centre of Pick 'n Pay's ISP. Configuration was done on the router and switch to allow traffic to reach the high-availability cluster, and connectivity tests were done to the servers, as well as access lists applied to restrict unwanted traffic to the servers.

Several scenarios were tested to find out how to most effectively route mail to the servers before sending it to the corporate network. Policy based routing was the first option tested, but failed due to a limitation on Cisco's IOS. The next option was to try NAT on the router, but again this failed, and it was decided to look at the servers themselves to try and find a solution. The only solution using the routers was one involving stopping traffic at the primary mx and thus forcing it through the secondary being the DSPAM servers where it would then be relayed to the primary.

Testing it was a challenge, but with the use of access-lists on the router we were able to test the configuration without interfering with production e-mail and systems. Testing was done via a approved change-control system during specified hours. A 'trial by fire' test happened when a power

grid upgrade was done to the building housing Pick 'n Pay's server room and power had to be shut down. Mail was spooled on the DSPAM servers until connectivity was restored and then successfully delivered to Pick 'n Pay's primary e-mail server.

An additional suggestion by one of the system administrators was the use of a SBL (Spamhaus 2004). The Spamhaus Block List ("SBL") can be used by almost all modern mail servers, by setting the mail server's anti-spam DNSBL feature (sometimes called "Blacklist DNS Servers" or "RBL servers") to query sbl.spamhaus.org. This was done on the remote DSPAM servers, and an immediate improvement in the amount of spam being rejected was noticed. This also lightened the load on the servers since a large quantity of spam was being rejected before even being processed by DSPAM.

3.6 Production

Before the system could be put in to production, a few additional tasks had to be taken care of that needs mentioning in this paper. First was training for the help desk staff on the DSPAM interface. Time and again, companies roll out new systems and solutions without adequately training the help desk staff in supporting the applications. Like any new system, there was going to be teething problems in the beginning, and training help desk staff to properly support the solution would limit the amount of calls logged, and make acceptance of the new solution a more pleasant experience for the users. A very simple help page was also published on the corporate intranet explaining how the solution worked via screen-shots, and this enabled help desk staff to easily diagnose problems users might be having with the new system.

Like many other corporate companies, Pick 'n Pay also limits the amount of web-browsing that staff can do in a certain amount of time. Staff are allocated a limit of 30mb web-browsing traffic per week. It would have been unfair to deduct checking their DSPAM quarantine inboxes from this quota, thus a alternate solution was put in to place. This was accomplished via a simple firewall rule routing traffic to the DSPAM servers on a different port, directly to the destination IP addresses. A simple link was then done on the intranet to the DSPAM servers. Users could simply click on the link 'DSPAM Quarantine Mail', and it would take them straight to the remote DSPAM server web interface, where they would have to log on with their user-name and password to view their quarantined e-mail.

3.7 Results

Quantifying the results of the project is pretty simple and straight forward. DSPAM has a statistics pages that provides information on the amount of real e-mail versus spam that users on the system receive. In some cases, at time of writing this paper, certain users had 2800 e-mails and 1100 of those have been spam messages. DSPAM also provides analysis graphs of the amount of spam versus real e-mails that it processes in a 24hr and 14day period as well as graphs per individual user to see their spam vs. normal mail ratio.

4. Conclusion

The nature of spam is such that one solution will never be the magic silver bullet to stop it. The sole purpose of this paper was to show how Pick 'n Pay Information Systems went about finding and implementing a open-source solution to fight spam and for this solution to be yet another layer of defense in a ever escalating battle between spammers and large corporates. With the proper investigation, planning, testing and implementation, open-source solutions in a corporate environment can work successfully.

5. **Glossary of Terms**

5.1 – MX Record - A DNS Mail Exchanger resource record that specifies where mail for a domain name should be delivered. You can have multiple MX records for a single domain name, ranked in preference order.

5.2 – ISP - An Internet Service Provider is a business which provides connectivity to the Internet. It provides you with the ability to send and receive Internet e-mail, browse the World Wide Web and download files from Internet servers. Internet Service Providers often offer other Internet-related services such as web-site design and hosting.

5.3 – DMZ - De-Militarized Zone, a no-man's land between the Internet and the internal network. This zone is NOT in the internal network, but is NOT widely open on the Internet. A firewall or a router usually protects this zone with network traffic filtering capabilities.

5.4 – Cisco IOS - Cisco system software that provides common functionality, scalability, and security for all products under the CiscoFusion architecture. Cisco IOS is a CLI that allows centralized, integrated, and automated installation and management of internetworks while ensuring support for a wide variety of protocols, media, services, and platforms.

5.5 SBL - The Spamhaus Block List (SBL) is a real-time database of IP addresses of spam-sources, including known spammers, spam gangs, spam operations and spam support services.

6. References

6.1 Falk J.D. (1998) The Net Abuse FAQ
http://www.cybernothing.org/faqs/net-abuse-faq.html

6.2 Glass B. (2000) Stopping Spam and Malware with Open Source
http://www.brettglass.com/spam/paper.html

6.3 Trend Micro (2004) Antivirus, Content Filtering, and Optional Anti-Spam Technology for the Messaging Gateway
http://www.trendmicro.com/NR/rdonlyres/36F69040-4586-4F5A-AE03-835DB692A987/12326/WP04IMSS55_040811US.pdf

6.4 Zdziarski J.A. (2004) Bayesian Noise Reduction: Contextual Symmetry Logic Utilizing Pattern Consistency Analysis
http://bnr.nuclearelephant.com/BNR%20LNCS.pdf

6.5 Zdziarski J.A. (2004) Concept Identification using Chained Tokens
http://www.nuclearelephant.com/papers/chained.html

6.6 Zdziarski J.A. (2004) External Inoculation Theory
http://www.nuclearelephant.com/papers/external_inoculation.txt

6.7 BMC Patrol
http://www.bmc.com/products/proddocview/0,2832,19052_19429_26305_8705,00.html

6.8 Spamhaus (2004) A guide to effective spam filtering
http://www.spamhaus.org/effective_filtering.htm