Solido Spam Filter Technology

Written August 2008 by Kasper J. Jeppesen, Software Developer at Solido Systems

Solido Spam Filter Technology (SSFT) is a stand alone filter that can be integrated into a wide variety of email services. It delivers unparalleled filtering quality and throughput performance on any Java capable UNIX platform. This document will introduce the reader to some of the technologies that make up SSFT. For further details you can request an NDA version of this document that is more in depth regarding the actual filters and their implementations.

Introduction

SSFT aims specifically at classifying pure spam mails–unsolicited bulk mail with the purpose of selling a product. Thus, we do not train our filter to detect viruses, phishing attempts or valid bulk emailing lists. Instead, we suggest that SSFT be used in combination with an anti- virus product such as ClamAv [1] to provide a full email filtering solution.

SSFT is currently in use at several larger Danish organizations that filter email for a wide range of international customers. We also employ the technology ourselves in our sister company, Armada Hosting. This gives us daily, real-world experience that we believe is needed to truly understand the challenges posed by running a large scale email service.

The rest of this document is divided into sections which describe the actual filter, training and maintenance of the filter, integration with SMTP[2] services, real world performance benchmarks, a small note on installation and administration, and information about sales and licensing.

---

[1] Clam AV is a free open source anti virus product available at clamav.net

[2] SMTP - Simple Mail Transfer Protocol is the commonly used standard to transfer email between servers

Filtering

When evaluating wether or not an email is spam, the things you can examine can be divided into two categories: context and content. Context covers the metadata related to the delivery of the email, e.g., the time of delivery and the IP address of the delivering server. Content covers the actual contents of the email. IP black lists such as Spamhaus[3] can be used to evaluate the context of the email by looking up the sending server's IP. Filtering technologies such as Bayesian text classification evaluate the content of the email.

The first part of the filtering process in SSFT is the initiation of the context filters such as DNS lists. These are typically dependent on third party resources, i.e., their execution time can vary wildly depending on the current load on those resources. We start these requests as a background process whose results can be polled when the rest of our filters have completed their work. In instances where some of the resources are responding very slowly, their results can be discarded while still maintaining a high spam detection rate and a high mail throughput.

Besides using standard DNS lists such as Spamhaus, we also employ our own non-binary IP reputation list. This list is continuously updated from all of our current spamfilter installations, based on results from our content based filters. It is also possible to add your own, third party DNS lists to the SSFT filtering process as long as they adhere to the same informal standard such as that used by Spamhaus.

Before any of the SSFT content filters begin their work, the email is preprocessed to ensure that different content filters don't end up performing the same work on the raw mail data.
This preprocessing includes a general attempt to parse the mime format of the mail, decode any attached images and separate the actual text content from html tags. Finally, all raw text parts are run through our custom high performing Hidden Markov Model based deobfuscation filter which detects and corrects obfuscation attempts. The result of this final preprocessing filter is not only a clean input for the rest of our text based filters, but also a filter in it's own right, which will affect the final result if it detects any clear obfuscation attempts in the processed data.

The primary text filtering in SSFT is done by a stack of three different text filters. A naive bayesian filters and two logistic regression filters. Due to the disjointness of the false positives set of each individual filter, the combined classification will have a much lower misclassification rate than the individual filters would on their own.

Although a big part of spam today is being completely custom generated for each email sent out, there are still big waves now and then that are similar enough that they can be caught quickly and effectively by a text based finger print filter. For this purpose, we fingerprint the text contents of emails using a very fault tolerant and proprietary transformation algorithm.

The final part of the SSFT filtering process is to examine any images that were attached to the email. The image filters currently in use fall into three separate groups: fingerprint, OCR[4] and heuristic.

---

[3] Spamhaus is a set of public anti spam DNS lists available at www.spamhaus.org

[4] OCR - Optical Character Recognition, different techniques used to recognize written/printed text

Fingerprint filters attempt to detect features in images that allow them to recognize the same group of images even when they are being dynamically created with big variances. We currently have several custom fingerprint algorithms in use that are each able to withstand different types of variance in the inspected images with almost no false positives.

The OCR filters in SSFT use different techniques to do full or partial OCR in order to recognize messages written in the images. These filters are able to deal with both changing fonts and sizes as well as "bouncing" letters inorder to detect messages written purely in images.

The final image filter group are heuristic filters that look for specific features of images. These are not in general use but instead represent more of a toolbox approach which allows us to quickly add a new feature detection filter that can detect images we are otherwise unable to detect. They can be thought of as a custom per-image type fingerprint filter system.

Once all filters have completed their evaluation of the email, the final result is calculated by using a small logistical regression filter that is trained along with each dataset. This ensures that the result of each filter is only trusted to the point of its actual performance. Negative changes in filter performance after training on a new set of emails will automatically be regulated before it can have an effect on the actual filter results.

Training

Unlike many other spam filters, daily updates to SSFT contain both code and data. This allows us to push new code out to all installations without the need for any local maintenance work. Having the ability to push out new code within moments of leaving our test center allows us to fight back fast and efficiently when encountering new types of spam.

Updates to the filter are pushed automatically from our servers to your local installation. The updates are hot swappable, ensuring that the daily updates never interrupt the mail flow.

All training for SSFT updates is done several times daily at Solido Systems. It is possible, but not necessary, for users to contribute to our datasets. This contribution can happen both as reports of misclassified mails as well as as automatic statistics about the types of emails being received from different IP addresses.

Integration

SSFT runs as a standalone daemon which communicates with clients over TCP/IP connections. Several well defined protocols are available for custom integration directly with the daemon. However, there are also several ready to use integration options included in the package: http, milter[5], spamassasin[6] replacement and spamassasin plugin.

The fastest and most direct way to create a new custom integration with the filter is through a simple text-based protocol. Connections to this service can be held persistently and using this

---

[5] Milter is a mail filter plugin architecture. See www.milter.org for more information

6 SpamAssassin is a commonly used open source anti spam filter available at spamassassin.apache.org

interface, a mail can be filtered either by streaming the mail data through the connected socket or by sending the path to a local file that contains the raw mail.

The http integration option is provided through a full http interface in the filter–effectively making the filter an easy to use web service. You simply send the email to the filter as a post request and get the result in the content body returned. This makes it possible to use standard libraries and tools such as curl to quickly integrate the service into almost any environment.

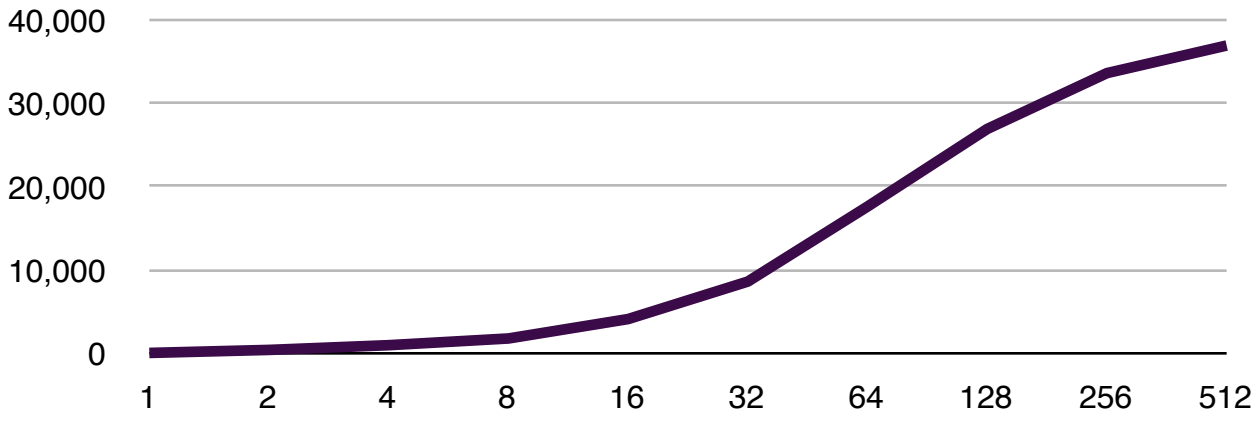
For those customers who are already running spamassasin, we have developed both a full set of spamassasin drop in replacement scripts as well as a single rule that can be used to embed SSFT in an existing spamassasin setup. This makes it very easy and risk-free to test out SSFT in your current environment.

Performance

SSFT has been highly optimized to provide maximum throughput on multicore hardware running multiple spam filtering processes in parallel. Each mail may take anywhere from 2 - 100ms to filter depending on the number of filters necessary and the performance of external third-party services such as DNS lists. However, as long as the filter is fed enough emails in parallel, the overall throughput will always stay very high.

There will always be a tradeoff between the quality of the filtering and the speed at which it can be done. While the Solido filtering technology in general performs well enough to allow you to use the full set of filtering technologies available, it can sometimes be beneficial to turn down the quality a bit to gain performance. This can be done on the fly–even while the filter is running.

The following performance results for SSFT were the result of running load tests in our own production environment. We have developed a record-playback solution using our SMTP proxy which allows us to record real world mail flow and play it back under controlled circumstances in a repeatable way. The following graph shows the results of stress testing on a quad core[7] Dell PowerEdge 1950 III using 200000 emails recorded in June, 2008 during peak hours at Armada Hosting. The horizontal axis show the number emails being filtered concurrently while the vertical shows the result as the number of emails filtered per minute.



[7] Quad Core Xeon E5420 2.5Ghz, 2 x 15k 73gb SAS drives, Perc 6I controller, 4gb 667mhz fb RAM

In order to allow the Java engine to "warm up"–i.e., JIT compile classes, cache DNS requests, etc–the test was done in two phases. The first phase consisted of restarting the filter and running the first 10,000 messages through without measuring. The second phase consisted of starting the measurements. We believe this approach best simulates a sustained running environment.

The best results–just over 33,000 messages per minute–were consistently achieved by running 256 filtering processes in parallel. Running with 1 ,4 and 16 threads–as available in our home, small business and enterprise licenses–resulted in about 200, 1000 and 4000 messages per minute. Be aware that the high end performance numbers are very dependent on a high performing DNS server setup in close network proximity to your SSFT installation. It is also important to note that these numbers only represent the actual spam filtering throughput. To achieve these performance levels, the SMTP service that access SSFT must be able to handle these high numbers of parallel operations as well.

Installation and Administration

SSFT requires a solid Java 1.6 environment. We test and run it under Redhat and MacOS X, but it should run under most UNIX environments. The installation is self contained in a single jar file which will store all files under /usr/local/solido/.

Once installed, all configuration is handled by editing a simple text based configuration file and restarting the service. It is generally not necessary to do any post install configuration, however. When the filter is running, it is also possible to telnet into a simple console interface that allows you to see the status of filters and emails which are currently being processed as well as basic statistical information.

The spamfilter logs its work to a simple text based log file which can be auto rotated daily, weekly or monthly.

Contact Information

For more information regarding sales or to request further information, contact us at:

Solido Systems

att: Christer Hasse
Troejborgvej 74
8200 Aarhus N
Denmark

tlf: +45 70 279 179
+45 22 482 828

email: info@solidosystems.com