

Technology behind SpamCheetah spam buster

Gayatri Hitech

August 9, 2009



Gayatri Hitech

<http://gayatri-hitech.com>



The OpenBSD appliance company!

Abstract

Spam control has traditionally been looked upon as a problem involving probabilistic mathematics, an exercise in linguistics and even in message digest computation. Of course math alone is not enough to combat a real life problem. All these approaches have their merits but it will be nice if we could combine this intelligence with some lateral thinking.

Spam control is as much a problem in the realm of TCP/IP networking or security as it is with mathematics. The typical tug of war between attackers and hackers in the security world is not very different from the way spammers adapt to spam control techniques. Hence the above mentioned academic approach may not always work. We need to strike the problem at the root. This paper discusses a technique known as OpenBSD greylisting that not only deals with spam effectively but also works with minuscule CPU, memory and consumes the least amount of bandwidth resources. We will see how we can turn away spammers from sending mail to us and even hurt them in the process.

Contents

1	Technical Summary	4
2	Unique Value Proposition	5
3	Under the hood	7
3.1	The software side	7
4	The Greylisting approach	8

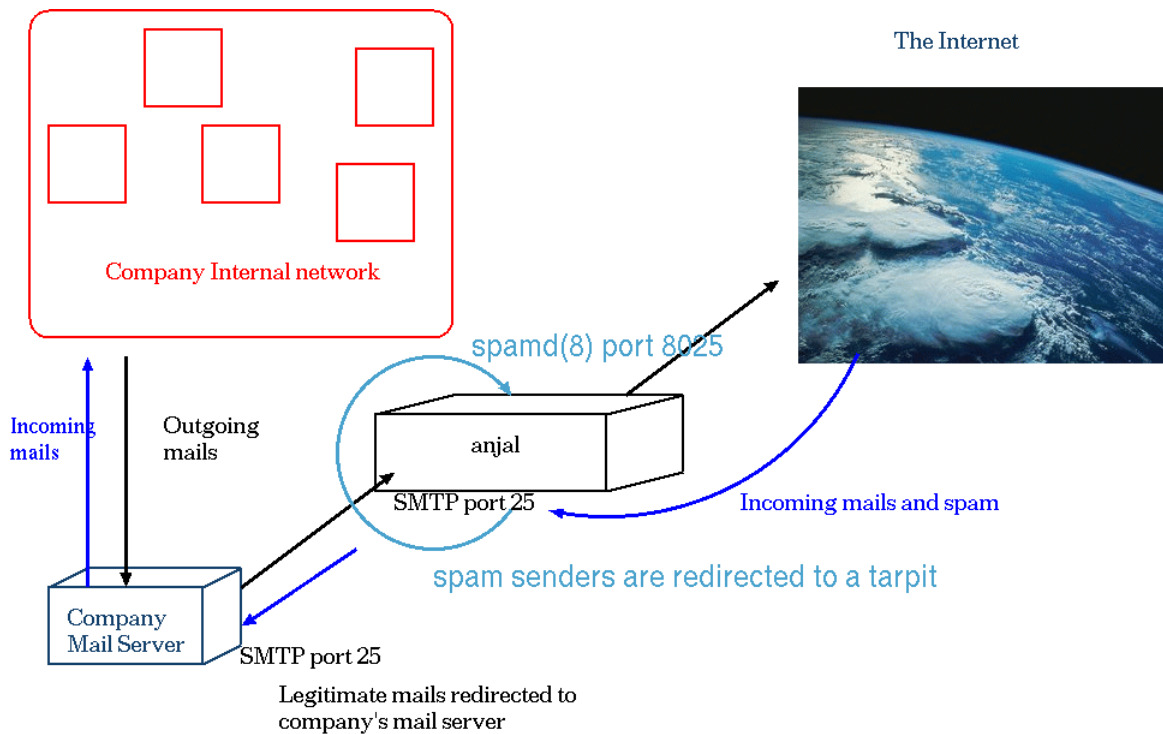
1 Technical Summary

SpamCheetah does spam control in a fundamentally different way. Though greylisting has been around for a long time now it has not been implemented in tandem with other techniques and in a proper manner to extract maximum leverage.

This is what SpamCheetah attempts to do.

Here is a schematic to explain how SpamCheetah works.

Figure 1: spamd architecture



The firewall that works in SpamCheetah redirects e-mail traffic depending on three parameters -

- sending IP address (From IP)
- envelope sender (who sends you mail?)
- envelope recipient (who is mail addressed to?)

If the above '3 tuple' are seen for the first time then the mail sender is subjected to the torturous SPAMD filtering (running on port 8025 above). There is a phenomenon called 'initial stuttering' that happens here. Instead of talking at full speed the MTA accepts mail one character at a time. This will piss off spammers and many go away. But legitimate senders have just one mail to send. Moreover they have to be RFC compliant. So they survive the test.

Once this process is completed, any subsequent mails from this sending IP address is assumed to be legitimate and they directly talk to the company mail server.

This in a nutshell is what my antispam solution does.

2 Unique Value Proposition

Internet e-mail has existed for quite some time now. In fact e-mail continues to be one of the most commonly used applications on the Internet. It has become an important business tool. The criticality of e-mail infrastructure for any business is obvious in today's world.

Spam however continues to dog us though spamming as a method to advertise and sell online was discovered much later in the evolution of Internet mail.

A variety of spam control techniques have been attempted by the smartest minds on the planet. And spammers have also been fairly good at circumventing even the best spam protection tools available today.

But along with containing spam with various levels of success we also face another important issue.

We lose legitimate mails occasionally if the spam control mechanism's internal clockwork goes haywire. In fact no one can guarantee you that you will never lose a legitimate mail.

Not all mails are lost due to spam control techniques however. But content scanning is a highly risky business. It is in the very nature of the math involved. Nothing much can be done over there.

We need something else.

We need a better tool or at least combine this technique with still better defences.

It is always nice to have more tools in our armoury especially since the adversary is well aware of the latest tools we possess.

In this arms race like in the case of cryptography and computer security, we have to attack at the level of human psychology.

In crypto, the keylength of encryption is made only as long as necessary to thwart a dedicated cracker. The cost incurred in cracking an encryption algorithm/security mechanism should be more than the value that one hopes to derive from the cost incurred.

Remember, it is a question of evolution and change. We can never know when the adversary will get the better of us. As our tools evolve so does his.

Spam control as a problem can be compared with computer security since we only need to make it unattractive for a spammer to offload his junk on us. Granted, he can always 'break our system'. But ultimately it is about the cost involved, it is about the motivation of the individual.

If we make it unattractive for the spammer to deliver his junk and at the same time ensure that the legitimate senders do not get affected in a big way then we are set.

Greylisting is a method to delay the delivery of e-mail since we force the mail sender to retry at a later point in time. Legitimate mail senders survive this acid test. But spammers don't.

Why?

Their business model does not account for the cost incurred retrying every mail. Theirs is a volume business. So what if you don't accept his mail? He can send it to thousand other unsuspecting victims.

A spammer does not have any particular interest in delivering his message to you alone.

Whereas a legitimate mail sender is very different. Moreover spammers run automated mail sender programs called as 'botnets' that do the sending.

Botnets are not written by overbright people but still they are written to ensure that they can thwart the defences of most of the spam control techniques.

But their motivation is limited to making money. If something does not make business sense they will not do it. This is the fact that we bank on for our defence.

3 Under the hood

3.1 The software side

Spam control has to invariably fall under one of the following categories.

1. Bayesian filtering and contextual analysis
2. Heuristical filtering based on known keywords/bad words
3. CRM114 Markovian chain based filtering (related to a)
4. Vipul's razor approach of DCC (Distributed checksum computation) with manual interference - gmail uses this heavily
5. Greylisting to stop spam right at the MTA level
6. IP address blacklisting and e-mail address whitelisting
7. TMDA - cure worse than the disease (Only approved senders can send mail)
8. RBL lists , spamhaus (politically sensitive spam control techniques)
9. SPF from Microsoft (not a bad idea per se) but does not work well

Most of these techniques are based on content scanning/filtering and actually reading e-mails with a computer.

Since this is an activity that requires a high end CPU and memory, spam control software and virus scanning software typically end up grinding your machines to a halt or even slow down your legitimate e-mails.

Also there is the very scary possibility of losing e-mails due to false positives.

My product 'SpamCheetah' uses the technique called OpenBSD greylisting. This is a very smart way to combat spam since it is stopped right at the MTA level. Since this never reads e-mail it is also very fast and highly efficient. It is impossible to get a false positive here though the first mail from a domain will experience a delay.

Basically greylisting forces mail servers to be RFC 2821 compliant and retry mails until the receiving site is ready. This also has an added advantage of hurting spammers sometimes and also stopping the spam that is meant for some other sites.

And you don't waste your storage space and bandwidth receiving spam first and then rejecting them.

4 The Greylisting approach

Greylisting is a relatively old concept and is well understood by many including spammers. But we take it to a totally different level by adding a lot more clever techniques that ensure that the spammer cannot get past us without significant effort on his part.

Let us see how we achieve that in detail below.

Anyway as a bonus this also stops all sorts of irritating malware like viruses, Trojans, worms and other annoyances.

Spam control is really not that hard if you understand how spammers do their job.

There is a lot of difference between the way spammers send out mail and the way legitimate mailers send.

Spam sending botnets (which are programs) send out e-mails in bulk. And their business is really dependent on volume. Even if one person in a million clicks on a p0rn site or is stupid enough to opt for some pleasure enhancement pill or some such thing they are fine.

This is precisely what we use for controlling spam by using OpenBSD greylisting.

Here are some even more interesting features.

- No false positives at all since this technique does not differentiate between e-mails, only senders
- Spammers actually get hurt by this. So if everyone used OpenBSD greylisting then spammers would go out of business.
- Since greylisting pushes back mail to the spammer, we do not accept spam even at an MTA level. This means that your bandwidth bills and storage space is saved.

E-mail sending is a two way process involving the sender and the receiver. The sender cannot send faster than the receiver says he can receive. It is this fact that we exploit in order to hurt spammers.

We create a tarpit for the spammer's botnet such that they get stuck in sending. They never get to deliver the mail and we keep wasting the spammer's resources. This is how we hurt spammers. Legitimate senders don't bother since they have only one mail to send.

The fact that we do not allow the spam to even enter our network saves us the trouble that the spammer intended to cause. And since we never read a single line of e-mail content there is no way we can ever wrongly classify a legitimate mail as spam.

We keep track of spam sender IP addresses and also refer to trusted up-to-date global databases for spamming botnets for tarpitting.

Our strategy of not reading mail content speeds up legitimate mail processing unlike content scanning/statistical filtering solutions.

Every human generated mail message (including mailing list mails) are ham. So there is no question of the subjectivity of what we call spam.

This device does one job and does it well. It does only spam control and does it well. It can handle a very high mail traffic due to the absence of content scanning. In addition to being itself capable of handling heavy load due to the intrinsic design and spam control technique, it can also scale and provide load balancing/failover using a protocol by name CARP. This comes in handy should you have hardware/software failure.

The company is also committed to providing the best technical support possible since all our products are aimed at 'plug in it, let it go' model.

Which is to say that it will require zero maintenance or baby sitting on your part. This is in contrast with content scanning based spam filtering solutions that are the order of the day that not only slow down your legitimate mail but also leads to false positives. You no longer have to sit and scan quarantined mails to decide whether to bounce, delete or accept your mail.

There are other reasons why we think our technology is a real breakthrough.

- There is no idea of false positives in our technology
- It does not slow down legitimate mail traffic
- Since it stops spam even before it gets a chance to get into the network, it saves you precious bandwidth and storage space
- Spammers actually get hurt by this, so if more people used our technology then there would be less spam on the Internet to deal with
- This technique can also be used to minimize viruses, worms and other malwares that float on the Internet in a transient fashion