# Silent Packet Dropping With Stockade

Adam Black
Centre for Advanced Internet Architectures. Technical Report 070424A
Swinburne University of Technology
Melbourne, Australia
adamblack@swin.edu.au

**Abstract- Stockade is a network-layer spam-mitigation package designed to prevent spammers from consuming excess bandwidth on a mail server. This report contains various tests performed on Stockade which assess how silently dropping packets affects spam sending patterns. The results concluded that when Stockade was active on the mail server and configured to drop packets silently using Ipfw [1], the rate at which clients could send spam was reduced significantly.**

## I. Introduction

This technical report has been prepared to inform the reader about several experiments conducted on Stockade 0.2 [2] to assess the Deny `IpfwRejectAction` property (available in Stockade's configuration file.)

Stockade is a network-layer spam-mitigation package for FreeBSD [3], designed to work in conjunction with mail filtering programs, such as SpamAssassin [4] and SpamBayes [5]. Stockade filters packets at the TCP layer and can prevent spam e-mails from consuming excess bandwidth on a mail server. Stockade uses probabilistic rejection [6] to determine whether an incoming SMTP connection should be accepted or rejected.

A feature of Stockade is the ability to automatically rehabilitate clients who have previously been registered as spammers. This rehabilitation feature means that an IP address registered as a spammer with a rejection probability of 1.0 will eventually be allowed to send e-mails without any chance of being blocked, given they do not send spam during the rehabilitation period. As a result a host who is constantly flooding a mail server with spam will have few of the connection attempts accepted, depending on the spammers sending rate. The rehabilitation period is dependent on the `MetricDecayFactor`, which determines how much time must elapse prior to the reject probability of a client halving.

Section II contains a brief introduction into Stockade, explains some of the properties available in Stockade's configuration file and covers some of the terminology used in this report.

Section III contains the main experiment conducted on Stockade using the `Deny IpfwRejectAction` setting. The purpose of this experiment was to determine if silent packing discarding is beneficial compared to the reset or reject methods, explained in Section II.

The report is concluded in Section IV and references are listed at the end of this document.

## II. Background On Stockade

Section IIA explains some of the technical terms used within this report. Section IIB outlines the basic usage of Stockade which includes examples of how to register spammers. IIC Explains the rehabilitation feature of Stockade and IID mentions the various Ipfw blocking techniques implemented in Stockade.

### A. Terminology

#### 1) Blacklist

Unconditionally block an IP address or range from SMTP communication with the server. A blacklisted host has a rejection probability of 1.0

#### 2) Greylist

Probabilistically block SMTP communication from an IP address/range. A greylisted host has a rejection probability greater than zero, but less than one.

#### 3) Spam Metric / Reject Probability

- Probability that a particular IP address is a spammer.
- Probability that an incoming SMTP session will be rejected.

### B. Basic Stockade Usage

Stockade provides an executable file and a Perl module which allows the user to block spammers. The usage is outlined below:

Using command line executable:

```
register_spam(tag, ip_addr, metric)
```

Using Perl module:

```
registerSpam(tag, ip_addr, metric)
```

The `tag` argument is a spam classification identifier and can be used by Stockade to normalise metrics from SpamAssassin for example.

The `ip_addr` can be specified in the form: a.b.c.d which will block a single IP address. It can also be expressed as a subnet range in the form: a.b.c.d/n, where n is the netmask prefix integer in the range 1 to 32. Note that a netmask prefix of 32 will specify a single host to block.

The `metric` is the initial rejection probability to be applied to the IP address or range.

### C. Rehabilitation and the Configurable Parameters

A feature of Stockade (known as automatic rehabilitation) is the ability for blacklisted and greylisted IP addresses to eventually become completely unblocked for SMTP communication with the mail server.

The rehabilitation parameters I will discuss include the following:

- RehabilitationHeartbeatInterval
- RehabilitationIterationsPerHeartbeat
- UpdateTickTime
- MetricDecayFactor
- MinDropProbability
- MaxDropProbability

Rehabilitation is implemented by reducing each senders drop probability according to an exponentially decaying function with the half life determined by the MetricDecayFactor property. Simplified, this means that after the MetricDecayFactor time has elapsed the probability of dropping new SMTP session halves.

When the spam metric is greater than the MaxDropProbability the IP address is blacklisted by Stockade and when the spam metric is less than the MinDropProbability the IP entry is unblacklisted by Stockade.

Invocations of the rehabilitation algorithm are performed sequentially on each IP address entry at the RehabilitationHeatbeatInterval. At every heatbeat multiple IP address entries can be rehabilitated, this is determined by the RehabilitationIterationsPerHeartbeat property.

The UpdateTickTime is the minimum number of seconds between rehabilitation calculations

### D. *Blocking Techniques*

When using Ipfw rule-based filtering there are three methods of denying a client communication with a mail server. Stockade lets you configure which method will be used via the IpfwRejectAction property. The available options and their descriptions are shown below:

- Deny - Drop packets silently
- Reject - Send an ICMP unreachable IP packet.
- Reset - Send a RST TCP packet.

The Deny method does not send any response to a blocked sender, which can cause the client's MTA to timeout waiting for a reply. Using the deny reject action will save upstream bandwidth by not sending replies to blocked clients. It is also more discreet than the Reject or Reset methods because it doesn't reveal any information to the spammer about why the e-mail exchange failed.

IpfwKeepState is another property of Stockade. When keep-state is turned on and a spammer's initial SYN attempt is blocked by stockade any subsequent SYN retries to the server are rejected unconditionally until the keep-state time expires.

During these experiments Ipfw was used in the Deny mode of operation with IpfwKeepState enabled.

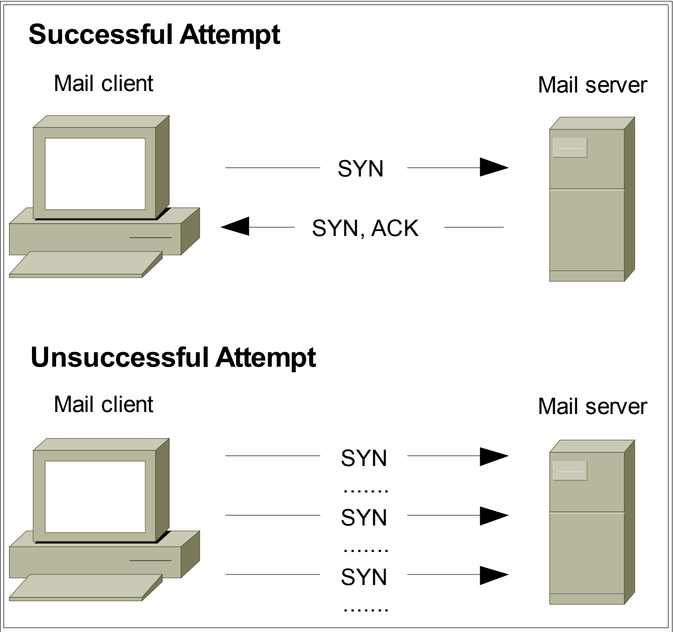

Figure 1: TCP conversation between mail client and mail server using Deny IpfwRejectAction

## III. Testing Deny Reject Action

### A. *Aim*

To determine how the Deny IpfwRejectAction property effects the sending rate of spammers and measure the difference in spam load on a server with and without Stockade.

### B. *Equipment*

- 2 x PC’s running FreeBSD 6.2 connected directly over a Gigabit connection. They will be referred to as the mail client and mail server.
- During the experiments the mail host used a modified version of smtpclient 1.0.0 (from FreeBSD 6.2 ports collection) which was capable of sending e-mails bound to alias IP addresses.
- The server used the following packages during the experiments:
  - sendmail 8.13.8.
  - ipfw (as with FreeBSD 6.2 release).
  - stockade 0.2.
  - procmail 3.22 [7].
- The client was configured with 20 IP addresses ranging from 192.168.0.200 to 192.168.0.219.
- The server was configured with a single IP address, 192.168.0.1.

### C. *Method*

In this experiment there were 16 ham senders and 4 spam senders e-mailing a server for one hour. The ham e-mails were sent serialised from each designated ham IP address in sequential order, while the spam was sent in parallel by each individual spammer process. This meant there was a 1:4 ratio of ham:spam being sent to the server assuming no interference from Stockade. The ham and spam sending processes were implemented as scripts, there was one script for sending e-mails from the ham addresses, and four scripts for each individual spammer. Each process sent e-mails to the server with an approximate one second delay between sending attempts. The configuration is shown in Figure 2.

The mail server used Procmail to determine if a sender was a spammer or not, by cross-referencing the source IP address of each incoming e-mail against a file containing the four IP addresses of predetermined spammers. If a spammer was detected they were blacklisted by Stockade. This process is demonstrated in Figure 3.

Stockade and Ipfw were initialised on the mail server. A `MetricDecayFactor` of 300 seconds was used, the `IpfwRejectAction` was set to `Deny` and IpfwKeepState was turned on.

A second test was run under the same sending conditions, but without Stockade and Procmail enabled on the mail server.

The five Python scripts returned individual logs containing the following information:

- The timestamp that each e-mail sending attempt occurred.
- Whether or not the e-mail was sent successfully.
- The status of the sender, either a hammer or spammer.

The individual logs were concatenated together and analysed to produce aggregate statistics which are shown in Section IIID.

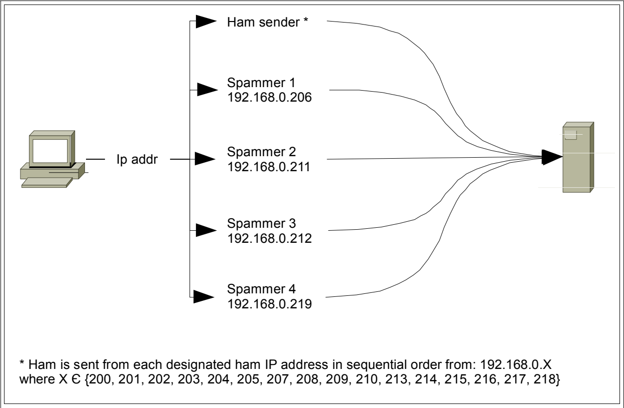
Ham sender *
Spammer 1
192.168.0.206
Ip addr
Spammer 2
192.168.0.211
Spammer 3
192.168.0.212
Spammer 4
192.168.0.219
* Ham is sent from each designated ham IP address in sequential order from: 192.168.0.X
where X Є {200, 201, 202, 203, 204, 205, 207, 208, 209, 210, 213, 214, 215, 216, 217, 218}

Figure 2: Sending behaviour of mail client

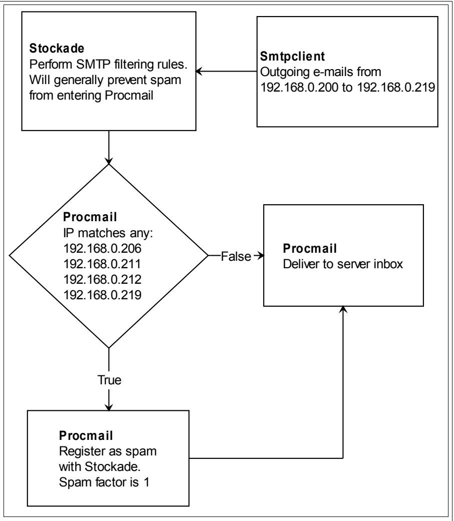
Stockade
Perform SMTP filtering rules. Will generally prevent spam from entering Procmail
Smtpclient
Outgoing e-mails from 192.168.0.200 to 192.168.0.219
Procmail
IP matches any:
192.168.0.206
192.168.0.211
192.168.0.212
192.168.0.219
False
Procmail
Deliver to server inbox
True
Procmail
Register as spam with Stockade.
Spam factor is 1

Figure 3: Procmail and Stockade overview

### D. *Results*

The numeric results from the two tests are shown in Table 1 and 2. Some graphical representations are displayed in Figure 4 and 5 which have been analysed and explained at the end of the Results section.

| Condition | #No. E-mails | Avg rate (msg/sec) |
|---|---|---|
| Total ham attempts | 3499 | 0.97 |
| Total spam attempts | 280 | 0.08 |
| Successful ham | 3499 | 0.97 |
| Successful spam | 89 | 0.02 |

Table 1: Ham and spam sending attempts over one hour with Stockade

| Condition | #No. E-mails | Avg rate (msg/sec) |
|---|---|---|
| Total/Successful ham attempts | 3500 | 0.97 |
| Total/Successful spam attempts | 14000 | 3.89 |

Table 2: Ham and spam sending attempts over one hour without Stockade

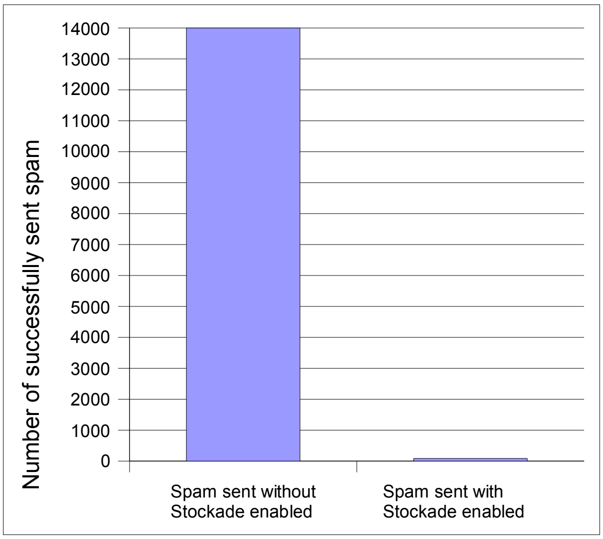

Figure 4: Quantity of spam sent to the server

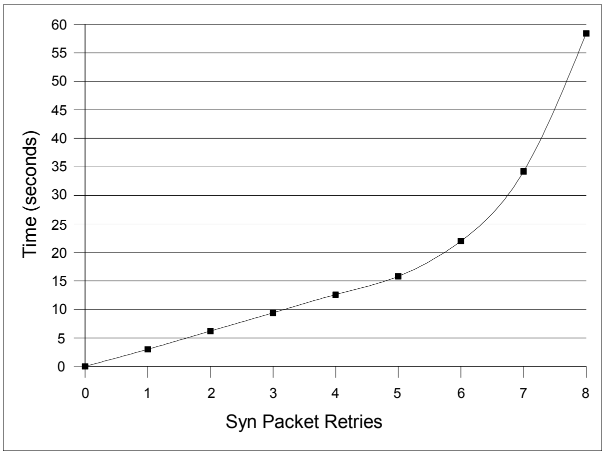

Figure 5: Syn packet retry interval over one minute

From the results we can calculate that Stockade reduced the amount of spam successfully reaching the mail server by 99.3%. Although 89 of 280 spam e-mails were successfully sent when Stockade was active, this was because Stockade reduced the sending rate of the spammers by causing their mail transfer software to hang while awaiting a SYN response from the server. This slow down between sending attempts gave sufficient time for the spammer's reject probabilities to rehabilitate.

Figure 5 shows a spammer resending SYN packets because they had been blocked by Stockade on each connection attempt. Note that after one minute of unsuccessful SYN attempts smtpclient timed out. Remember that `IpfwKeepState` was enabled in Stockade's configuration, which meant that after a spammer's initial SMTP connection attempt was blocked by Stockade, any consequent SYN retries would unconditionally be blocked for a lock-down period of 20 seconds. After this lock-down period expired and the spammer changed state from blacklisted to greylisted, any of the subsequent SYN retries (prior to smtpclient timing out) would be probabilistically accepted/rejected.

Table 3 demonstrates the amount of time required for a spammer who was blacklisted to be reduced to a certain spam metric.

| Spam Metric | Time (secs) |
|---|---|
| 0.95 | 22.2 |
| 0.9 | 45.6 |
| 0.85 | 70.3 |
| 0.8 | 96.6 |

Table 3: Rejection probability rehabilitation with MetricDecayFactor of 300 seconds

Table 2 shows that the average e-mailing rates of each spam and ham source was approximately 1 msg/sec per source. This is because the e-mail transmission code and the implicit one second sleep command caused the sending interval to be slightly greater than one.

### E. *Future work*

To more comprehensively test Stockade it would be useful to run a similar set tests while varying the parameters listed below:

- `MetricDecayFactor`
- `IpfwKeepState`
- `IpfwKeepState` hold-time (kernel parameter, not a property of Stockade)

## IV. Conclusion

Stockade is a spam filtering package designed for FreeBSD with a unique auto-rehabiliation feature which removes the need for manual blacklist maintenance.

The Stockade configuration file has a parameter named `IpfwRejectAction`, which determines how packets are dropped by Ipfw. We set this parameter to `Deny` and observed the effect this had on incoming ham and spam e-mails.

We concluded from the results that using the `Deny` `IpfwRejectAction` to silently drop packets decreased the amount of spam on the server by 99.3%. This was also able to significantly reduce the sending rate of spammers because their sending scripts would have to wait for either a SYN response from the server or a time-out before the next e-mail attempt could occur.

Due to the reduction in possible spam attempts, and the discreetness of silent blocking, the `Deny` mode is an efficient way of dealing with spammers.

## References

[1] "FreeBSD Handbook Chapter 27 Firewalls", http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipfw.html (viewed 2 April 2007)

[2] M. Robb, G. Armitage, Stockade, http://caia.swin.edu.au/stockade/ (viewed 2 April 2007)

[3] "The FreeBSD Project", http://www.freebsd.org (viewed 2 April 2007)

[4] SpamAssassin, http://spamassassin.apache.org (viewed 2 April 2007)

[5] SpamBayes, http://spambayes.sourceforge.net (viewed 2 April 2007)

[6] M. Tran , "Mitigating Email Spam by Statistical Rejection of TCP Connections Using Recent Sender History", Australian Telecommunication Networks and Application Conference 2006, Dec 2006 (http://caia.swin.edu.au/pubs/ATNAC06/Tran_M2m.pdf) (viewed 2 April 2007)

[7] Procmail, http://www.procmail.org (viewed 2 April 2007)