Identifying Spammers in Outbound Email Systems

A minor thesis submitted in partial fulfilment of the requirements for the degree of Master of Computer Science

Martin François Foster
School of Computer Science and Information Technology
RMIT
Melbourne, Victoria, Australia
January 24, 2011

Declaration

This thesis contains work that has not been submitted previously, in whole or in part, for any other academic award and is solely my original research, except where acknowledged.

This work has been carried out since March 2010, under the supervision of Dr. Ibrahim Khalil.

Martin François Foster
School of Computer Science and Information Technology
RMIT
January 24, 2011

Acknowledgement

I would like to thank my supervisor, Dr. Ibrahim Khalil for his feedback throughout the process, and particularly his guidance in choosing a reasonable amount of data and metrics to analyse.

I thank my wife Nicole and daughter Stefanie for travelling to Canada in the final weeks of my write-up process, and our families to hosting them while there. Without this time to focus, this minor thesis would probably have never been completed.

Contents

List of Figures

List of Tables

Abstract

Large scale Service Provider email systems have always been targeted for potential exploitation by spammers because of these systems' capability to deliver huge volumes of their nefarious payload.

Today, most email systems decide whether to accept email from another system based the sender system's network reputation. Reputation provides Service Providers with the incentive to minimize spam originating from their network; allowing too much spam to be sent via their facilities is likely to result in a poor reputation, meaning other reputable service providers will reject their mail.

Compared to the volumes of research targeted to detecting spam in inbound mail flows, there has been relatively little work done in identifying spam in outbound mail flows. The research that has been done on outbound flows is difficult to apply to modern mail systems, in that it discounts either their complexity, has too much reliance on simple metrics such as volume, or provides mechanisms that are only suited for offline analysis – by which time it is too late to act.

Service Provider email system complexity must be accounted for in order to build a complete picture of the paths and transformations that affect messages on their way out of a mail system: typically crossing multiple servers and multiple different software packages. Old metrics such a message sending volume are likely to penalize the wrong party; with the advent of stringent anti-spam laws legitimate mailing lists tend to deliver a high volume of email to recipients desiring this content. Whereas spammers have moved away from sending high volumes from one host over a short amount of time – preferring to send low volumes from many hosts over a long period of time to net higher overall delivery and avoid or delay detection by service providers.

This thesis presents a novel mechanism to detect spammers in the outbound flows of Service Provider e-mail systems. It accounts for the complexity of such systems, can be used for near real time analysis, and uses the foreign destination system's response as a measure of the probability that the sender is a spammer.

Chapter 1

Introduction

Most research in the field of e-mail spam detection has focussed on inbound flows to an e-mail system. Inbound flows are defined as the messages a system under the control of one administrative group receives from other systems that are not under this group's control.

Less focus has been placed on vetting the outbound flows from e-mail systems. Outbound flows are defined as the messages that originate from subscribers of an e-mail system under the control of one administrative group destined for receivers on other systems that are not under this group's control.

Much of the prior work done in detecting spammer activity in outbound flows is difficult to apply to large scale service provider e-mail systems because it discounts their complexity. This thesis focuses on the large scale use case; the work in and raw data used by this thesis is based on the author's experience in designing and operating an Internet Service Provider (ISP) e-mail system servicing approximately 100,000 subscriber mailboxes. The author believes the methods used apply to any large scale e-mail hosts, such as ISPs, and email hosting services such as Google, Microsoft, Yahoo and Fastmail. Unless otherwise noted, all references in this thesis to e-mail systems imply large scale.

Where does the spam in outbound flows originate from?

Spam is defined as unsolicited commercial email (UCE) or unsolicited bulk email (UBE). Spam's goal is usually financial. Direct financial gain by generating revenue via the sale of a product or fraud. Indirect financial gain by stealing access to the target's resources, usually by intercepting their system credentials or convincing the target to install malicious software on their computers.

The malicious software angle is important. Because of the introduction of Anti-Spam legislation, the number of persons and organizations overtly spamming or providing Internet

access services to spammers has greatly reduced: what was previously frowned upon is now illegal. Coupled with the widespread adoption of network reputation services, almost no e-mail systems are operated to directly deliver spam anymore. Instead most is originated by computers that are running malicious software, usually as a part of a larger network of systems running similar malicious software under the control of one organization – a Botnet. These compromised accounts are responsible for the bulk of the spam being injected into e-mail service providers' outbound flows.

Operators of email systems are motivated to reduce spam in their outbound flows for reasons of deliverability, resource consumption, and reciprocal improvement.

Deliverability is the measure of how much legitimate e-mail sent by customers of a mail system gets received by its intended recipients. At the receiving system side, messages are first accepted into inbound flows based on the sender's reputation. An indication of reputation is provided by consulting one of the many Realtime Blocklists (RBLs), who indicate how likely a host on the Internet is to deliver spam based on past observed behaviour. If operators of an email system do not manage and control spammer's use of their systems, their reputation will degenerate in these RBLs until such point that few other systems will accept their mail. Customers will quickly cease to use an e-mail system with low deliverability due to poor reputation.

The financial cost of sending an e-mail is small [1]. However, if the overall volume of spam sent is not controlled, service providers will incur considerable infrastructure and staff costs. If a 10 server system is overprovisioned by 25% in order to managed high utilization peaks caused by queuing spam for delivery, then 2 servers worth of infrastructure are not justified by customer use or revenues. Personnel is more expensive than infrastructure; sending excess outbound spam requires more staff to process a system abuse reports, and to trace and manage compromised accounts.

The final motivator is reciprocal. Due to economies of scale, there is an ongoing trend for smaller companies and organisations to outsource their email hosting needs to larger providers. The net effect is a larger volume of mail is moving between relatively fewer e-mail systems. If these systems in turn better detect spammers in their outbound flows, they should eventually face lesser inbound flows to filter for spam.

1.1. Objective

The objective of this work is to identify spam in the outbound mail flows of large scale service provider mailsystems. This would provide means of reducing the volume of spam delivered to mailboxes, thus reducing the financial motives to spam.

1.2. Contribution

This thesis presents a novel mechanism to identify spammers in the outbound flows of Service Provider e-mail systems. It accounts for the complexity of such systems, can be used for near real time analysis, and uses the foreign destination system's response as a measure of the probability that the sender is a spammer.

1.3. Organization

The problem of identifying spammers in outbound e-mail system flows requires a good understanding of how electronic mail is routed from one system to another. Chapter two explains this, introduces the syslog data collection mechanism, and the intricacies in retracing the path that a message took through a high volume multiple server e-mail system.

The background chapter concludes with an overview of related works in determining e-mail sender reputation, assessing e-mail sender behaviour, and identifying outbound spam.

The third chapter presents the techniques and algorithms used to first retrace the paths taken by messages through the system, and analyse these paths for evidence that the source is spamming. The results demonstrate that using the receiving system's SMTP reply code alone is a good metric for identifying spammers; one that could be used to nearly halve the amount of outbound spam generated by an e-mail system.

Chapter 2

Background

The context in which large Service Providers operate must be understood in order for this thesis' objective to appear sensible.

2.1. Simplified Email System Overview

The design of large scale service provider e-mail systems is presented below. The network level protocol used to move mail between systems is the Simple Mail Transfer Protocol (SMTP), referenced in [2], with pertinent portions of its transactions defined later in this section.

2.2. Basic Functions

Stripped of all other features, an Email system has three functions as seen in figure 1. (1) it accepts mail on behalf of others, (2) it delivers mail to the messages' intended recipients, and (3) provides disconnected operation by queuing messages where the recipient is not currently available for a later re-delivery attempt.

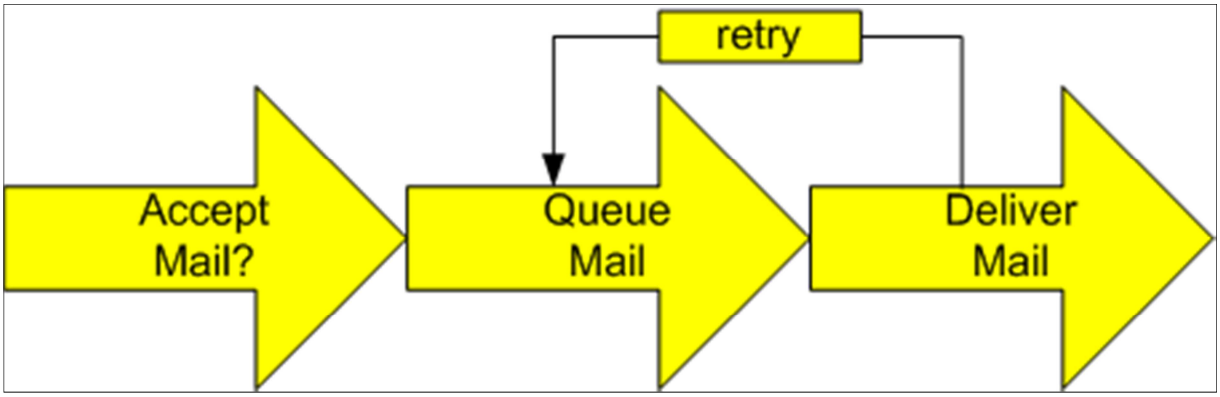

Figure 1: simple email system

2.3. Mail Flows

Email systems have two primary flows. Messages will take either an inbound or outbound path, depending on their origin & destination.

2.3.1. Inbound Flows

These messages are entering the mail system from other mail systems that are external to its administrative realm, or *foreign* to it. The foreign systems will have been directed to the local system by Domain Name System (DNS) Mail Exchanger (MX) records, or explicit administrator configured routing. Prior to accepting these messages, the local system will probably validate that the message's recipients are valid mailboxes.

2.3.2. Outbound Flows

Messages sent from subscribers to mailboxes in *foreign* systems. It is the undesirable mail originating from the service provider's subscribers that will impact the reputation of the outbound flows, which in turn will halt or delay delivery to foreign systems that consult and trust major Reputation Providers.

As previously mentioned, this thesis focuses on outbound flows.

2.3.3. Intelligence from Outbound Flows

Service providers have a considerable amount of intelligence relating to outbound mail flows that is not available for inbound flows. This is the information that will be used by the detection algorithm.

Key intelligence sources are:

- Availability of raw mail logs, and the ability to preserve these logs for baseline historical analysis.
- Ability to determine which customer sent a message: either the originating IP address, or the authenticated user name.

Using the above, service providers can exert far more fine-grained control on outbound flows than their inbound variant.

2.4. Simplified Email System Components

The opportunities, problems and constraints of a system are best understood by visualising its components.

Figure 2 enumerates the components of an email system that could be implemented by a service provider. Figure 3 overlays inbound and outbound mail flows over these components to demonstrate their areas of primary concern.

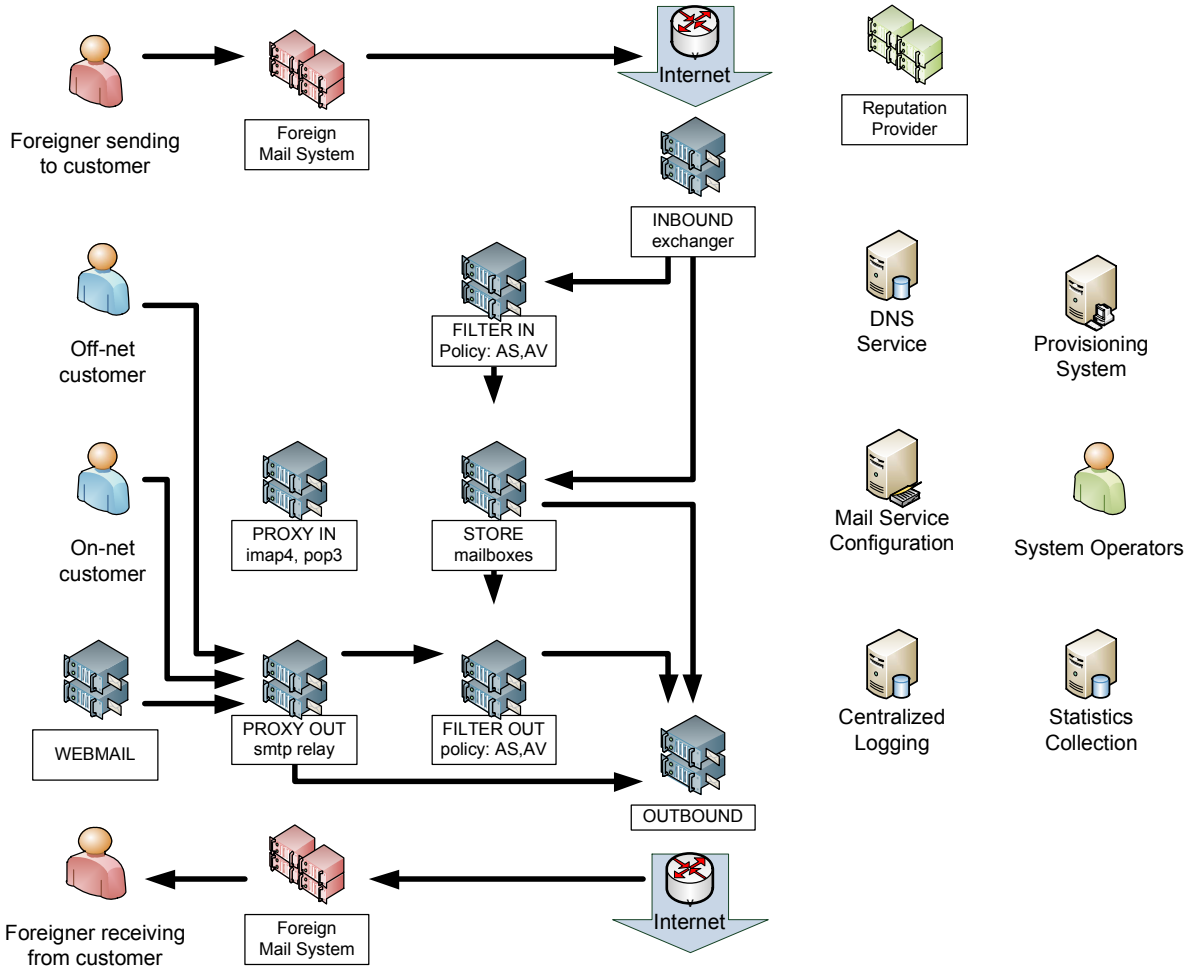

Figure 2: service provider Mail System Components and SMTP flows

2.4.1. Inbound Components

Referencing Figure 2, the components handling *inbound* mail flows are:

- Inbound Mail Exchangers, these receive mail from foreign systems
- Inbound Filtering, where anti-spam and anti-virus processing is applied by policy
- Mail Stores, contain the mailboxes that messages are delivered to
- Inbound Proxy, used to isolate the store from direct communication with customers. Proxies provide an abstraction layer to the mailboxes, which can be used to implement load balancing, reroute traffic during mail store maintenance, and provide the ability to rewrite or transform login credentials.
- Webmail, provides a web-based mail client. Often a prime target for attack or abuse, hence its isolation.

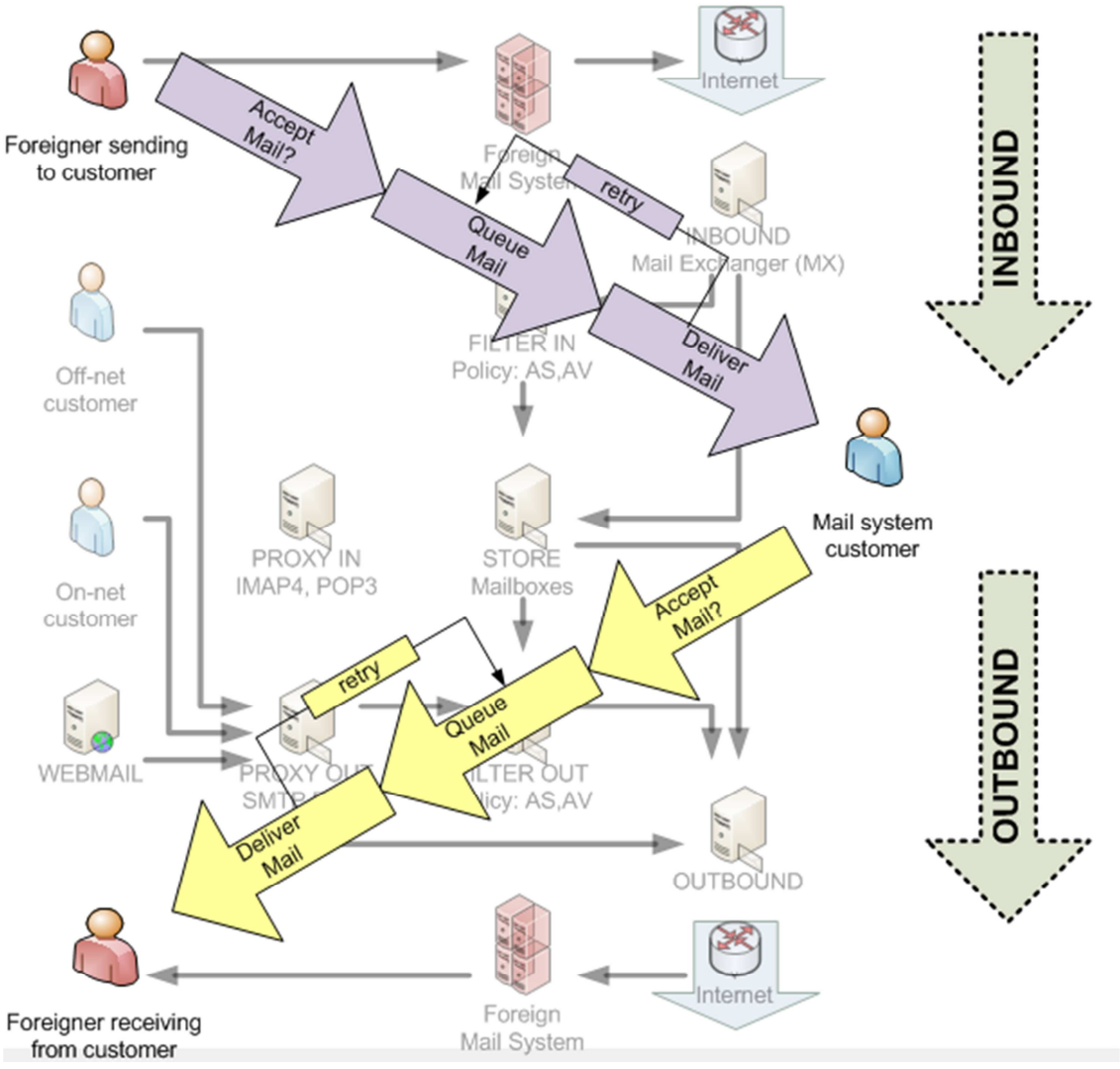

Figure 3: mail System Components and in/out flows

2.4.2. Outbound Components

Referencing Figure 2, the components handling *outbound* mail flows are:

- Outbound Proxy, the entry point for accepting messages that subscribers wish to deliver.
- Outbound Filtering, where outbound anti-spam and anti-virus is applied by policy.
- Outbound Queue system, the servers that will make the direct connection to the foreign destination servers. Also the only place in the mail system where messages should queue for delivery. Reducing queuing points reduces bottlenecks.

2.5. Large Scale introduces Complexity

Due to volume, the functions of a mailsystem are usually segregated to different servers. Redundant servers are deployed to keep the service available despite component failures.

This complicates mail flow analysis. The analysis provided in this thesis requires an aggregated view of the messages each subscriber submits to the outbound flows. The path a message takes through the email system will have to be rebuilt from information output to system logs as it traverses the many servers and daemons along its way.

2.6. Simple Mail Transport Protocol (SMTP)

This section provides a brief overview of the Simple Mail Transport Protocol (SMTP).

Proposed in 1982, RFC821 [3] standardized the client-server message exchange, and RFC822 [4] standardized the format of the messages. These have since been obsoleted by RFC's 2821 [2] and 2822 [5], but these updates and clarifications have no material effect on this thesis.

2.6.1. SMTP Example

Figure 4 shows a complete SMTP transaction of a simple email being sent from a client (grey background) to a server (white background).

This paper focuses on actions at the envelope level, and SMTP response.

The actual message content, steps 3, 4, 5, 6 in Figure 4 is not of concern to this work. The structure of the message is dictated by RFC822 & 2822, the rest of the communication is RFC821/2821.

1. Connection	**$ telnet smtp.pacific.net.au 25**
	Trying 125.255.95.1...
	Connected to mail-pacific.pacific.net.au.
	Escape character is '^]'.
	220 mailproxy2.pacific.net.au ESMTP Gday mate
	helo netlog.net
	250 mailproxy2.pacific.net.au
2. Envelope	**mail from: <martin_foster@pacific.net.au>**
	250 2.1.0 Ok
	rcpt to: <sender@netlog.net>
	250 2.1.5 Ok
3. Message	**data**
	354 End data with <CR><LF>.<CR><LF>
4. Message header	**Subject: This is my test message**
	From: "Some Sender" <fake@donottrust.net>
	To: "Some Recipient" <sender@netlog.net>
5. Message body	**This is my message body.**
6. Message end	**.**
7. Server send	**250 2.0.0 Ok: queued as F2F9727412**
	quit
	221 2.0.0 Bye

Figure 4: a SMTP transaction

2.6.2. SMTP Envelope

The example in Figure 4 shows that a message's envelope can use different addressing than the message's body. Note the difference between the envelope's "mail from" and "rcpt to" instructions versus the message header's To: & From: lines. This distinction often comes as a surprise to persons not familiar with e-mail's protocols.

This is similar to paper mail – the envelope is responsible for the routing of the message, the message inside the envelope can contain whatever the sender wants it to. SMTP is the same, the envelope routes the message's payload.

2.6.3. SMTP Reply Codes

In Figure 4 the server responds to client commands with a reply code in the 200-299 range. These replies are defined in RFC821/2821. For the purposes of this work, we want to know three broad response categories to a given command:

- Accepted: 200-299 range.
- Failed temporarily: 400-499 range. This means that something went wrong in processing the command, but that this failure may not be permanent. The sender is encouraged to try again later. For example, if the destination system is too busy or unavailable, the sending system is likely to return a temporary failure and retry delivery for given period of time. Systems consulting a RBL when deciding whether or not to accept inbound messages from a remote system will usually give a temporary failure response code if the sending IP Address is blocked. This provides a mechanism in which to advise the sender and the sender's system's administrators as to the reasons for the rejection.
- Failed permanently: 500-599 range. This means that something went wrong in processing the command, and that this command as given will not be accepted. This sort of response is often given to the "rcpt to" envelope command if the given recipient does not exist on the system.

2.6.4. SMTP Relay

E-mail systems should not relay mail for anyone on the Internet. Relaying is the act of accepting a message from a customer into a system's outbound flows, and processing it for delivery. As previously mentioned, in large scale e-mail systems this processing often involves traversal of multiple servers, and queuing at the end of the chain for acceptance by foreign receivers.

Most systems use two criteria when deciding whether to relay. Either:

1. Client network address, or "relay by IP address". The client's network address is looked up, if it belongs to addresses that the email server will relay for, it is accepted. This is usually used by ISPs providing relay services to customers on their network.

2. Authentication. Covered in the next section, allows anyone with the proper credentials to send email.

Figure 4 demonstrates relay by IP address – no authentication is done.

For the purposes of managing spam, the developing trend is to move away from client network address checks to authentication. There are a few reasons for this.

With network based relaying, all messages from a source will be accepted. For administrators, this makes tracking the source of an exploited system more difficult as many computers could be in use behind a given address. If the one address is blocked while investigating a problem, many people are affected. There is also a series of exploits[1] against networking equipment that allows attackers to exploit and use home & small business routers as SMTP relays that the downstream ISP email system will accept messages for. Relaying from authenticated accounts only and stopping the practice of relaying by IP address protects against this entire category of exploits.

With authentication, it is possible to provide unique credentials to each user. This makes identifying the exploited account and device a much simpler process. Authentication also enables users to relay via from their service provider's e-mail system regardless of which network or physical location they are in; a bonus as the number of portable devices capable of sending email continues to increase.

2.6.5. SMTP Smarthost Servers

The term "smarthost" is used to describe servers that whose primary purpose is to relay mail for other servers in a multi-server system. They are effectively relays as described in the section above, and often deployed in places where direct outbound access to the internet is prohibited. The term is frequently used in datacentre or campus deployments.

[1] Limited formal research in this field, but it is a behaviour observed by ISPs. See http://technocrat.net/d/2008/1/12/33775/

2.6.6. SMTP Authorization and Encryption

The transaction in Figure 4 is very simple. At the establishment of the connection, other capabilities are often supported by the receiving server. Important ones are:

- Session authorization. A sender can be authorized to relay messages via e-mail systems that would otherwise reject them via authentication. The authentication mechanisms that a particular server supports can be queried by the client, the most common types being a username + password combination.
- Channel encryption. A SMTP transaction can be begun on a clear text channel, and subsequently cyphered with the STARTTLS command.

2.7. E-mail System Logs

The analysis of individual message's flow through an e-mail system will be done via system logs. The generation of these logs is discussed here. The algorithms and analysis of the resulting logs is covered in Chapter 3.

2.7.1. Syslog Mechanism

The syslog ("system log") mechanism and protocol allows software to generate event logs when a given action occurs. Syslog is provided by the Linux operating systems used in this work. The concepts carry to Microsoft Windows Events [6] which are similar in function but different in implementation and wire-level protocol.

Syslog provides means to state the facility for which a given message is generated, such a mailsystem, scheduler, kernel – and the severity of the message, from debugging, informational, warning, error and critical error.

The mechanisms and protocols are fully described in RFCs 3164 [7], and 5424 [8]. For the purposes of this work, note that informational messages are generated for most steps in the processing of an e-mail message.

In a large scale e-mail system, the logs from all participating servers tend to be aggregated at a centralized server ("loghost") for analysis. The logs used by this work were collected from a loghost.

2.7.2. Outbound Flow Reconstructed from Logs

Figure 5 is the syslog representation of the SMTP transaction in Figure 4. Each line is numbered for quick reference.

This is the output provided by the Postfix Mail Transport Agent (MTA). There are other software packages available that can perform this task; if they are suitable for use in large scale e-mail systems they will log similar information.

In cross-referencing, the:

- client's IP address is on line 1.
- the postfix daemon assigns a queue identifier (`F2F9727412`) on line 2. This allows the message to be traced in the receiving server – the queue identifier is returned to the client at the end of the transaction in Figure 4.
- message-id is generated in line 3, this identifier should be globally unique for the message's entire delivery from sender to final recipient.
- envelope sender and recipient is given on lines 4 & 5.

The system logs also show that while the client's message was accepted by a server called "mailproxy2", it was subsequently handed off to another server "mailout1" within the mailsystem for queuing and delivery to the foreign recipient mailsystem. The postfix daemon running on "mailout1" issued a different queue identifier (`16E444C817F`) on line 6, but preserved the message-id on line 9.

The log messages:

1. 2009-05-22T02:02:59+00:00 mailproxy2.pacific.net.au postfix/smtpd[1081]: **connect from** unknown[203.100.230.80]
2. 2009-05-22T02:02:59+00:00 mailproxy2.pacific.net.au postfix/smtpd[1081]: **F2F9727412**: client=unknown[203.100.230.80]
3. 2009-05-22T02:03:00+00:00 mailproxy2.pacific.net.au postfix/cleanup[8950]: F2F9727412: **message-id=<4A157B2D.1030204@pacific.net.au>**
4. 2009-05-22T02:03:00+00:00 mailproxy2.pacific.net.au postfix/qmgr[19788]: F2F9727412: **from=<martin_foster@pacific.net.au>,** size=649, nrcpt=1 (queue active)
5. 2009-05-22T02:03:00+00:00 mailproxy2.pacific.net.au postfix/smtp[25662]: F2F9727412: **to=<sender@netlog.net>, relay=mailout.pacific.net.au[61.8.0.84]:25**, delay=0.16, delays=0.13/0/0/0.03, dsn=2.0.0, status=sent (250 2.0.0 Ok: **queued as 16E444C817F**)
6. 2009-05-22T02:03:00+00:00 mailout1.pacific.net.au postfix/smtpd[19765]: **16E444C817F**: client=mailproxy2.pacific.net.au[61.8.2.163]
7. 2009-05-22T02:03:00+00:00 mailproxy2.pacific.net.au postfix/qmgr[19788]: F2F9727412: removed
8. 2009-05-22T02:03:00+00:00 mailproxy2.pacific.net.au postfix/smtpd[1081]: disconnect from unknown[203.100.230.80]
9. 2009-05-22T02:03:00+00:00 mailout1.pacific.net.au postfix/cleanup[19769]: 16E444C817F: **message-id=4A157B2D.1030204@pacific.net.au**
10. 2009-05-22T02:03:00+00:00 mailout1.pacific.net.au postfix/qmgr[23206]: 16E444C817F: **from=<martin_foster@pacific.net.au>,** size=869, nrcpt=1 (queue active)
11. 2009-05-22T02:03:10+00:00 mailout1.pacific.net.au postfix/smtp[19651]: 16E444C817F: **to=<sender@netlog.net>, relay=mx.netlog.net[208.78.102.55]:25**, delay=11, delays=0.03/0/8.5/2.1, dsn=2.0.0, status=sent (250 2.0.0 Ok: **queued as 4CF462201AC**)
12. 2009-05-22T02:03:10+00:00 mailout1.pacific.net.au postfix/qmgr[23206]: 16E444C817F: removed

Figure 5: a SMTP transaction seen from system logs

2.8. Related Work

The author believes that using SMTP response codes to identify spammers in complex systems is a novel approach to managing the outbound spam volumes originating from an e-mail service provider's infrastructure.

There is a substantial body of work in the field of spam mitigation, machine analysis of behaviour, and inter-system reputation.

2.8.1. Reputation

Third parties have long acted as reputation brokers for communication between two other organizations.

2.8.1.1. The Email Context

The effectiveness, design and evolution of DNS based Realtime Blocklists (RBLs) as e-mail reputation systems was covered in 2007 by Alperovitch, Judge, and Krasser [9].

RBLs list IP addresses known to have been the source of spam, or URL known to have been seen in the content of spam. Because of their binary spam/not-spam tagging behaviour, effective RBLs have to update cautiously in order to minimize false positives. This cautious update process has been heralded as their principal weakness. The advent of rapidly changing spam sending botnets documented in [10] was seen as the demise of RBLs [11].

In response to this criticism, some reputation providers have begun offering multiple data sets that have differing goals and update velocities. For example MAPS offers a long lived RBL+ high assurance list and a rapidly adapting QIL list [12].

The author has empirical data suggesting that RBLs continue to eliminate over 90% of inbound spam flows with very low false positive. Other parties have reported numbers in the 80% range when applied to botnets alone [13]. The actual efficiency of RBLs will vary based on mail system size, mail system age (the longer a system has been around, the better it is known to spammers), and the number of domains whose email is hosted by this system.

There exists many RBL providers, the most popular ones listing known spam sources are Spamhaus [14], MAPS [12], and SpamCop [15]. The most popular list of URLs placed in spam message content is SURBL [16].

The existence of these reputation providers and their continued relevance is the basis for this research in spam sender identification: to avoid impacting mail system users that exhibit "good" behaviour while stopping or penalizing the bad.

2.8.1.2. Outside the Email Context: P2P

Research in Peer to Peer (P2P) networks have proposed reputation based mechanisms to ignore or exclude nodes that are known to offer decoy files: files that are something else than what they claim to be. This is essentially the spam of the P2P world.

The Credence system described in [17] and [18] implements a reputation system similar to email RBL providers – peers are rated by the positive and negative endorsements of others.

2.8.2. Behaviour

This work will draw on prior research into the behaviour of spam networks, legitimate senders, and spam senders when attempting to classify subscribers.

2.8.2.1. Network Behaviour of Spam

In 2006 Ramachadran [13] then in 2007 Gansterer [19] investigated the network level behaviour of spam sending networks. While there is evidence that these networks are adapting to advances in detection mechanisms, this research provides a good baseline for identifying a "standard" botnet.

2.8.2.2. Network Behaviour of Spam: HVS versus LVS

In 2008 Pathak, Hu, and Mao [20] operated a selective e-mail relay server to gain an updated understanding of how spammers were using misconfigured e-mail servers to relay mail. Their work is important because it confirmed system and network operator observations that spammers appeared to be changing their behaviour.

Earlier, spammers were generally observed sending high volumes of spam to misconfigured or compromised relay systems as soon as they were discovered. The authors codified this behaviour as High-Volume Spammers (HVS). Spammers knew that a short window of opportunity existed to use the system before its administrators noticed the volume and reacted to it, or RBL operators noticed the volumes and listed the server as a source of spam.

What the authors observed is that more e-mail was being sent by Low-Volume Spammers (LVS). Instead of sending a mass volume of email through the relay over a short period of

time, they would send a low volume of email from many different sources over an extended period of time – the goal being to avoid detection.

The authors also noted that some of the clients exhibiting LVS behaviour appeared to be working under the command of a centralized orchestrator: multiple members of a coordinated botnet.

2.8.2.3. Foreign System SMTP Reply Codes

The work done by Richard Clayton observing the logs of UK ISP Daemon Internet in 2004 [21] uses the same metric as this work does to determine the likelihood that a sender is a spammer: SMTP result codes provided by a foreign system during a delivery attempt. This topic is also discussed in this thesis in section 2.6.3. The difference with this work is in system complexity and volume.

Their work showed that a specific sender which exhibits a low overall rate of delivered messages is more likely to be a spammer. Spammers tend to use lists of email addresses with a higher percentage of invalid or no longer valid email addresses. These email addresses are usually rejected by the foreign system when a delivery attempt is made, with the rejection indicated within the SMTP protocol and logged to the system logs.

The service provider e-mail system described for Daemon Internet at the time used one type of server as a "smarthost" (see section 2.6.4) for relaying outgoing mail from customers. While there were many similarly configured system for redundancy, the path taken by messages in outbound flows was only one server "deep", a simplification that greatly eases log analysis because a message's traversal through the infrastructure does not have to be rebuilt from system logs to extract and correlate the source's connection on one server to the response codes returned by the foreign e-mail system as seen on another server.

The last major difference is that in 2004, the majority of spam was delivered using High Volume Spam (HVS) whereas today the bulk is Low Volume Spam (LVS).

2.8.2.4. Behaviour of Email Users

An observed fact about email user behaviour is that people's social networks tend to be reproduced in their email patterns.

In 2002, Pujol et al. [22] worked to identify social networks in generic transactions between agents. The research attempted to map peoples' social circles by analysing who they communicated with, and how often. While this research was not originally applied to e-mail communications, there is reason to believe it would be applicable because all e-mail transactions contain the required sender and recipient information.

Columbia University's Intrusion Detection System (IDS) lab's Email Mining Toolkit (EMT) described in [23], [24], and [25] proposes and implements multiple mechanisms for abnormality detection in sender behaviour. Metrics track volume behaviours in time, the development of expected recipients by building maps of social networks (cliques), and the identification of changes as a result of exploitation or takeover by a malicious sending agent.

A subset of the EMT known as the Profiling Email Toolkit (PET) was implemented in [25] for real-time analysis. It was restricted to acting at the mail client endpoint, and not directly on a large scale distributed mail system.

2.8.3. Outbound Spam

The last category of related work focussed on mechanisms to reduce outbound spam without taking into account user behaviour. [26] Proposes three mechanisms that could be implemented by large scale e-mail service providers, all relate to imposing a cost – either financial or computational – on the sending of a message.

The problem with these proposals is that to be effective cost-based mechanisms need to be accepted by a large set of organizations exchanging email. So far this has not happened as there are: (1) too many objections to putting a cost on a resource that is currently seen as free or near-free, (2) no one can agree on who would regulate or collect a financial "postage" fee for sending a message, (3) cost based mechanisms are biased against developing economies that have less financial and computational resources, and (4) it is now assumed that spammers would easily absorb any computational cost by using their existing spam sending networks. The disagreement on financial cost and non-effect of a computational cost appears to have halted development of all mechanisms proposed by [26].

Chapter 3

Architecture and Algorithm

In this work, in order to determine whether the messages that a subscriber delivers to an e-mail system are spam, we must trace the path taken through the system by the subscriber's messages.

The correlated result provides a complete view of the treatment of each message sent by a subscriber. Building this complete view for e-mail systems which have more than one server or one server type involved in relaying outbound e-mail is part of the novelty of this work.

In this section the challenges for building this complete view and how they were resolved are explained.

3.1. Data Source: E-mail System Logs

The data source analysed in this work originates from e-mail system logs generated by servers and the Postfix MTA application. In the background section, the syslog mechanism and protocol was introduced as a standard means of collecting information about the processing of emails as they traverse a system.

Note that while syslog is used here, the process would be similar for other logging or infrastructure tracing mechanisms. Examples of alternate technologies are Microsoft's Windows Events [6], and Facebook's Scribe [27].

3.1.1. E-mail System Log Collection

A central syslog host is used to aggregate the streams of syslog events emitted by all the servers in the outbound flow processing chain. The analysis system processes this centralized data.

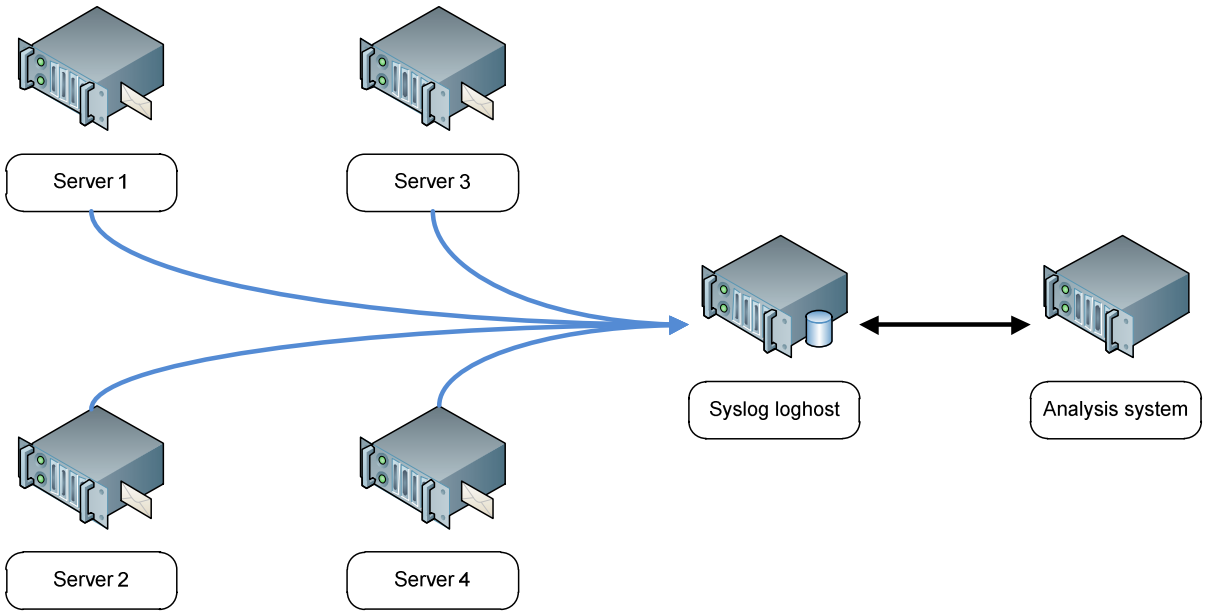

Figure 6: centralized syslog host

Figure 6 provides a graphical representation of the log aggregation process. The loghost writes its logs in a format that complies with the updated syslog RFC 5424 [8] published in 2009. This format is more reliably parsed from a text form than its predecessor RFC 3164 [7].

3.2. The Problem of Complexity

E-mail system complexity makes the task of building the complete view of a message traversing a system much more difficult than the single server case.

As the term implies, the simple case illustrated in Figure 7 has a single application (denoted as a daemon) running on a single server.

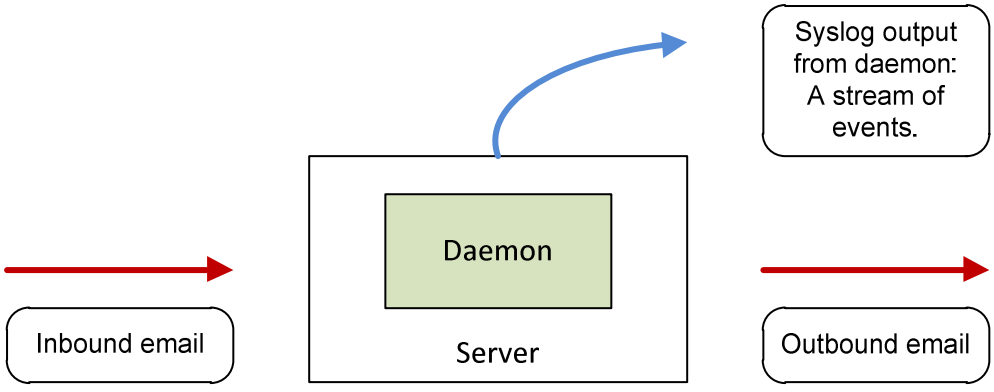

Figure 7: simple case, one server one daemon

The single server type still allows for many servers of the same type to be deployed to for reasons of redundancy or capacity. If many servers with the same functional configuration are deployed, it can be said that their operation is idempotent for any given recipient.

Complexity is introduced by using more than one application in processing mail, and having more than one functional processing stage.

3.2.1. Multiple applications

In mailsystems, it is common to have multiple applications or daemons process a message to get a result. This is illustrated in Figure 8, where a message would have to traverse three applications before exiting the server.

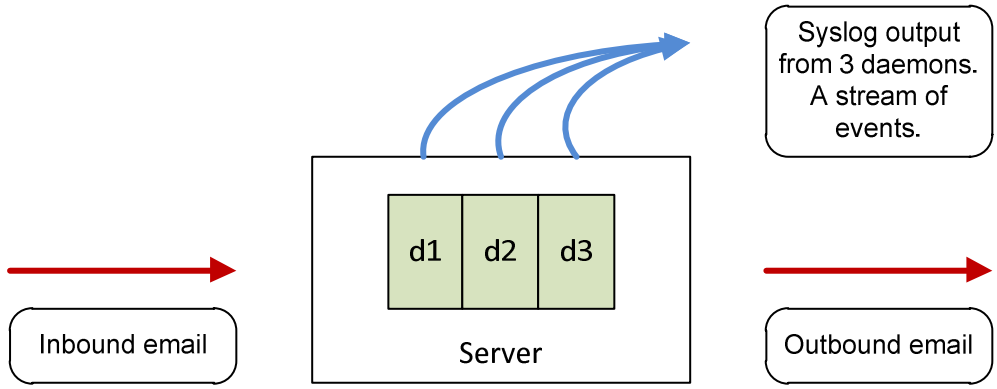

Figure 8: single server multiple application

An example of this is processing by anti-spam and anti-virus content analysis software. The first daemon is an MTA that accepts messages from the upstream system, the following daemons are content processors. The processed message is handed to the exit MTA for delivery to the next system in the chain.

This case is not tracked in this analysis, but would be a logical extension of the research.

3.2.2. Multiple Applications, Multiple Servers

For the analysis framework to be suitable to as many real world e-mail systems as possible, it must cope with the logs generated by e-mails as they traverse multiple servers, each potentially running multiple daemons.

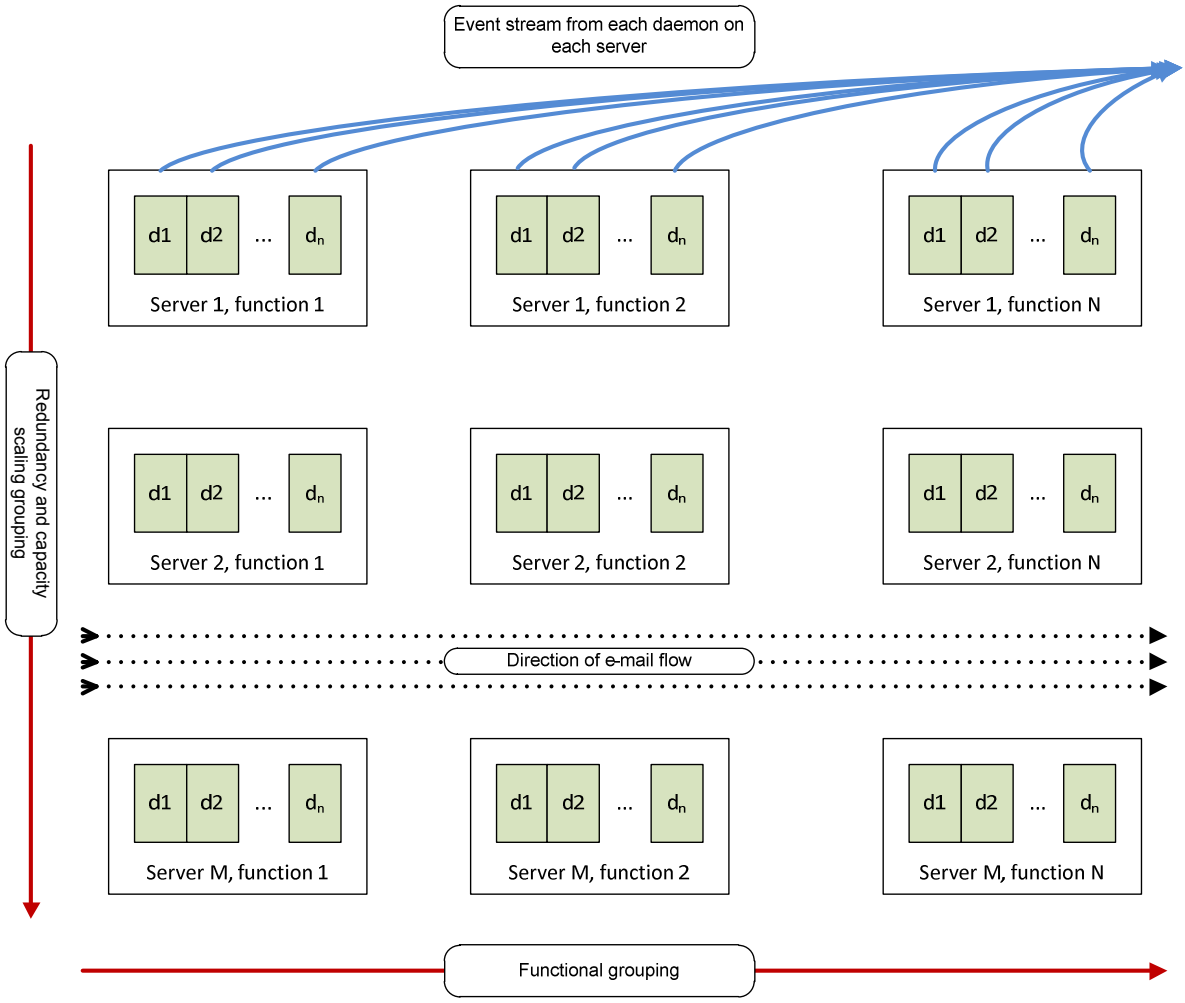

Figure 9: multiple applications, multiple servers

The general case is illustrated in Figure 9. Any e-mail entering a system can be processed by any number of applications for a given functional grouping. Functions are expressed in the horizontal axis, and functional replicas for throughput, redundancy, or datacentre diversity are expressed in the vertical axis.

The number of paths that an email can take when traversing a given system is computed by multiplying the number of functions N by number of servers per function M. This will vary if each function is not serviced by a constant number of servers.

3.2.3. E-Mail System under Test: Pacnet Internet

The actual system under test in this work is the outbound flow processing stage used by Pacnet Internet as shown in Figure 10.

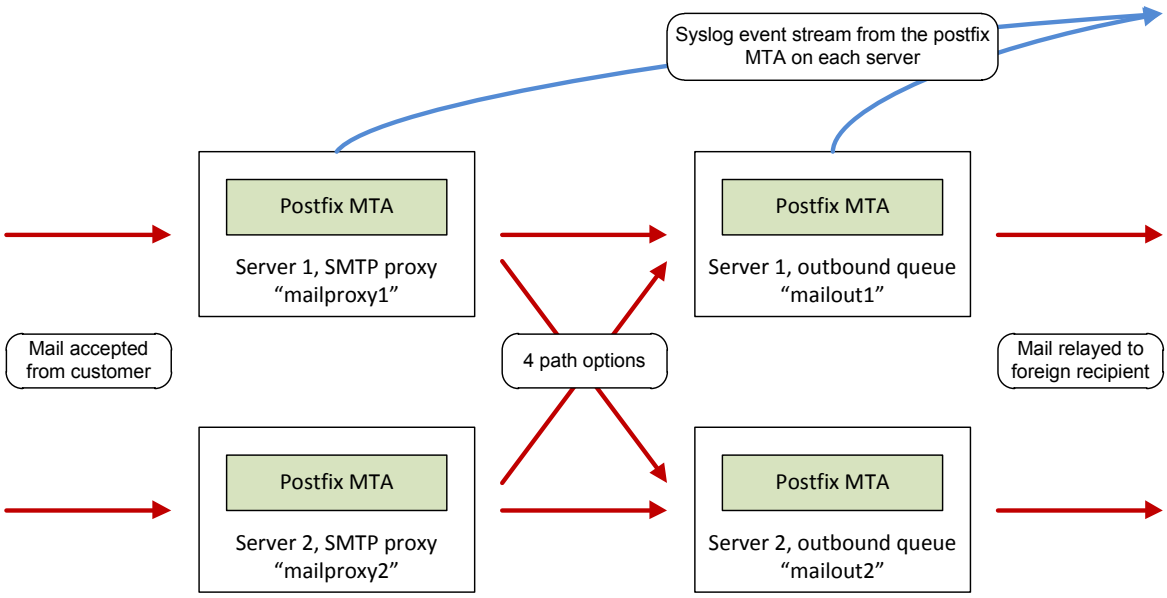

Figure 10: system under test, Pacnet Internet Outbound e-mail servers

Outbound messages are subject to two functional server types.

1. Front-end proxies that accept messages for relaying and validate that they are originating from customers. This validation is done using mechanism introduced in 2.6.4: the originator's IP address, or authentication.
2. Back-end mail queues handle the actual delivery to the recipient's e-mail system.

The functions are separated because the back end queuing process can take time to clear if there is an issue with delivering the message to the destination. Handling peak loads can also make the back-end systems less responsive. Other mail servers can tolerate this unresponsiveness and retry, but experience has shown that e-mail clients and the persons using them cannot. Hence the front end servers are kept lightly loaded to provide a low response latency.

From this design we can infer that in building the complete view of a message traversing the system, copies to each recipient will show two hops, using one of the four paths:

- Mailproxy1 to Mailout1
- Mailproxy1 to Mailout2
- Mailproxy2 to Mailout1
- Mailproxy2 to Mailout2

When there are multiple recipients to a message, each recipient's path taken through the system is not guaranteed to be the same.

3.3. Mail Log Parsing

The syslog data emitted from the e-mail system has to be parsed by the analysis system in order for interesting data to be extracted.

3.3.1. Postfix Syslog Information

The e-mail infrastructure that supplied the source data for this work runs the Postfix MTA [28]. Four syslog message types are needed to track the flow of messages:

1. **Acceptance**. Emitted when the software accepts a message for relay. The source address is provided, and if authentication was used the credentials are also logged.
2. **Message-ID**. Emitted when the globally unique message-ID is found or generated. The message ID may have been generated by the client or programs that have previously processed this message. In this case the message ID would be written to the message's headers, which the MTA can later extract and re-use for logging purposes.
3. **Sender**. Emitted when the software processes a message in the queue. The message's envelope sender is printed, along with other statistics such as the size of the message, and the number of recipients seen in this message. Note that the number of recipients seen cannot be trusted for accuracy, as prior servers may have split a message delivery to a larger number of recipients into parts.
4. **Recipient**. Emitted once the result of a delivery attempt to a recipient is known. Provides 3 possible states for this result, either:
    a. Sent, the delivery was successful. A SMTP reply code of 200-299 was received.
    b. Deferred, the delivery failed but may succeed in the future and so is re-queued for a future attempt. A SMTP reply code of 400-499 was received.
    c. Bounced, the delivery failed and will not be retried. A SMTP reply code of 500-599 was received.

Deferred messages exhibit a transient state. Delivery is retried until either a concrete response is received (accept or reject) from the recipient's mailservers, or all attempts to

deliver the message fail within a specified timeframe. By standard, this timeframe is 7 days, but in practice many operators reduce it to a lesser number such as 3 or 5 days to reduce the size of their outbound queues.

Only the two final states of sent (accepted) and bounced (rejected) are of interest to this work.

3.3.2. Parsing a Large Volume of Events

The volume of events to be processed in order to build the desired complete view of a sender's messages further constrains the solutions available to the problem.

Over the time the mail system under test was observed, an average of approximately 190,000,000 log messages were generated per week. This is a sustained average rate of 314 events per second, with peak levels many times this.

High end disk storage systems are required in order to have the input-output capacity to perform over 300 operations per second. Most syslog daemons batch writes into larger block in order to not impose high resource needs. For analysis it is important not to require high throughput and high cost storage systems that would incur a disk seek in order to read or write each event.

With this storage system constraint, the author decided to use an event stream processing system that operates in memory to handle the high volume first stage event parsing and aggregation.

3.4. Event Stream Processing

The syslog information emitted from an e-mail system is representable as an event stream. Each processing stage on each server generates an event. In order to build the desired complete view of what happens to a sender's message, all the events related to this message must be captured and sequenced.

The science of working with and reacting to streams of events is Event Stream Processing (ESP) and Complex Event Processing (CEP), and is covered in detail in [29] and [30]. For this work, the open source ESPER [31] component was used to correlate and extract the required Postfix Acceptance, Message-ID, Sender and Recipient logging messages.

3.4.1. Selecting Events in a Stream

The ESPER component uses Event Processing Language (EPL) to express the conditions under which events or groups are to be selected from a stream. EPL is a SQL-like language.

A simplified event stream is presented in Figure 11. It shows a variety of postfix (A)cceptance, (M)essage-ID, (S)ender, and (R)ecipient syslog messages being processed as events. For brevity, this combination will be referred as AMSR from here on in.

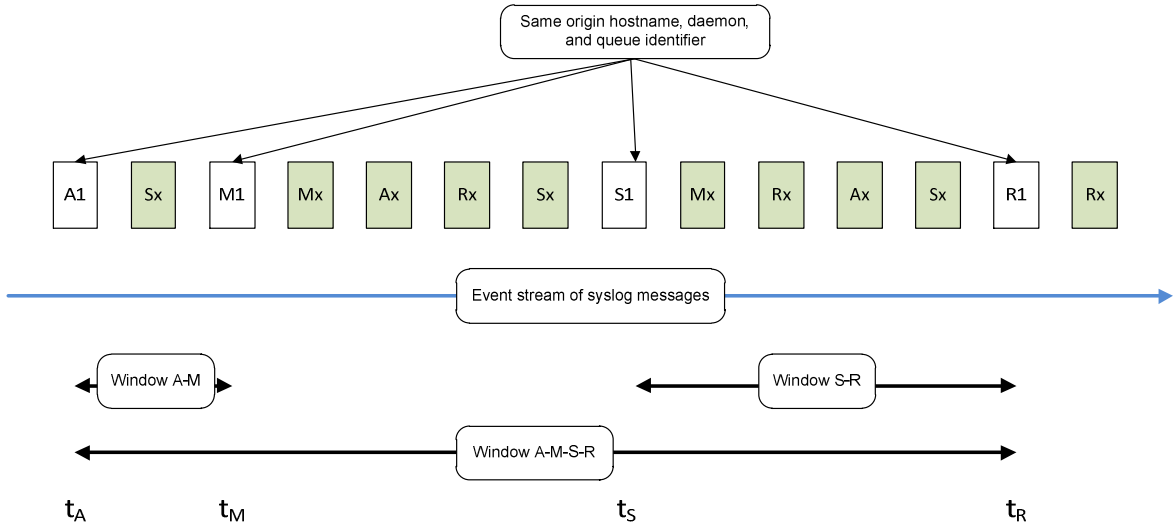

Figure 11: an event stream with postfix accept, message-id, sender and recipient events

When grouping processing on one host, the common data to aggregate on is the hostname, daemon, and queueID. Figure 12 shows an EQL statement capable of selecting the desired AMSR event from the event stream. Since the daemon is always postfix for this input data, the daemon component is ignored.

```
select pa, pm, pf, pt
from PostfixAccept:ext_timed(pa.timeLong, 10 minutes) as pa,
PostfixMessageID:ext_timed(pm.timeLong, 10 minutes) as pm,
PostfixSender:ext_timed(pf.timeLong, 10 minutes) as pf,
PostfixRecipient:ext_timed(pt.timeLong, 10 minutes) as pt
where pm.queueID = pa.queueID and
pm.queueID = pf.queueID and
pm.queueID = pt.queueID and
pm.hostname = pa.hostname and
pm.hostname = pf.hostname and
pm.hostname = pt.hostname
```

Figure 12: an Event Query Language (EQL) statement for selecting AMSR events

There are a few things to note in this process and EQL statement. First, the input syslog events must be pre-parsed and injected as Plain Old Java Objects (POJO), which makes their attributes accessible to EQL. Second, the queueIDs and hostnames are matched as previously mentioned. Third, the time window; each event's object representation has attributes for the hostname, daemon, queueID, and syslog timestamp. The syslog timestamp, which is stored as an attribute named "timeLong" is used instead of the ESPER injection time so that the actual time the syslog message was generated is used as a basis for comparison. Using this external clocking ("ext_timed") allows the system to catch up after a fault or processing peak and still produce the same output.

Also note that in practice there is no guarantee that these events will be logged in order – an M can be recorded before an A, and an R before an S. This is further reason to use ESP: it allows analysis systems to ignore arrival ordering, instead leaving ESPER to report all matching events within a selected time window.

3.4.2. Window Sizes

Next the proper window size must be selected. Event Stream Processors output events matched at the end of a given window. This means that when choosing a window size, it must be made large enough to observe an event of interest, but not so large that it is too late

to act by the time the end of the window is reached. Larger window sizes also require more memory resources to maintain, offering a second practical limit to their size. A ten minute window was chosen for this work.

Figure 11 is marked with the times at which the individual AMSR events arrive as t_A, t_M, t_S and t_R. "A" & "M" events usually follow one another closely once a server decides to accept a message for delivery. "S" events do not begin until the e-mail queue is processed.

The time between the "S" and "R" events varies depending on how busy the sending & receiving servers are, the network bandwidth, and the size of the message. For the purposes of this work, it was assumed that the entire A, M, S and R events were emitted within 10 minutes of one another.

This covers most cases, with the exception of long run delivery re-try cycles. These occurred in a small minority of cases, and are excluded for simplicity. They could be re-introduced as SR events that do not have a matching AMSR combined event within the given time window.

3.5. Reconstructing the Message Path

The AMSR matching described in the sections above converts a high volume of individual events to a much lower volume of per-server events. Per-server because AMSR events match on hostname and queueID, both are unique at the server level only.

To rebuild the entire path that a message has taken, AMSR events must be aggregated as a multiple server transaction. This is done by matching the globally unique message-ID, with ordering done by timestamp. Timestamp ordering assumes that the servers have synchronized clocks, say by using Network Time Protocol (NTP).

Some merging of information is done when suitable AMSR sets are found. Authentication information if it exists will only be found on the system that first received the message from the customer. Subsequent servers within the email system relay by IP and so will not have this authentication information. Hence it is preserved if present when merging sets.

3.6. Identifying Spammers by SMTP Reply Code

This work uses one metric to determine whether a sender's messages are likely to be spam, the SMTP response code. This is the same metric used by Clayton in 2004 ([21] and prior

work section 2.8.2.3) on Daemon Internet (UK) email, but applied to a more complex set of servers, and with the intent of detecting both high and low volume spam.

For each reconstructed message, and each recipient in this message, the SMTP reply code of interest is the code seen on the last path a message took.

The sender is assumed to be spamming if in time its messages have a too high percentage of undelivered mail. The mail is undelivered usually because the e-mail address does not exist at the recipient system, either because it is mistyped, no longer exists or never existed. Because spammers harvest a lot of their e-mail addresses, a significant proportion of their lists have bad or old e-mail address data.

3.7. Reconstructed Message View

Bringing together all the concepts in the above section yields the reconstructed view of messages traversing the e-mail system's outbound flows.

```
1  SENDER[10]: user1@isp1,43,43,0,0,0
2  message: 000001cba976$0fb13d60$2f13b820$@net.au
3  sender: user1@isp1
4  srcIP: 125.255.XX.YY
5  auth: NO
6  size: 1885701
7  nrcpt: 3
8  recipient: user2@hunterlink.net.au, path size: 2
9      path[0]: mailproxy2.pacific.net.au[B455C27803] user1@isp1-
   >user2@hunterlink.net.au status=SENT tries=1 destIP=61.8.0.85
10     path[1]: mailout2.pacific.net.au[007782AA1DC] user1@isp1-
   >user2@hunterlink.net.au status=SENT tries=1 destIP=61.8.0.83
11 recipient: user3@bigpond.com, path size: 2
12     path[0]: mailproxy2.pacific.net.au[B455C27803] user1@isp1-
   >user3@bigpond.com status=SENT tries=1 destIP=61.8.0.85
13     path[1]: mailout2.pacific.net.au[007782AA1DC] user1@isp1-
   >user3@bigpond.com status=SENT tries=1 destIP=61.9.189.122
14 recipient: user4@companyA, path size: 2
15     path[0]: mailproxy2.pacific.net.au[B455C27803] user1@isp1-
   >user4@companyA status=SENT tries=1 destIP=61.8.0.85
16     path[1]: mailout2.pacific.net.au[007782AA1DC] user1@isp1-
   >user4@companyA status=SENT tries=1 destIP=203.0.AA.BB
```

Figure 13: reconstructed message

Figure 13 shows a sample reconstructed view of a message generated by the analysis software written for this work. A few interesting parts of the view data:

- Line 1, for this envelope sender, the number of messages seen (43) and delivered successfully (43). A perfect delivery rate is unlikely to be spam.
- Line 2, The globally unique message ID
- Line 7, the message has 3 recipients, for each recipient the 2-hop path between a mailproxy and a mailout system is seen. Complete with the IP of and status reported by the receiving mail server (SENT is accepted).

Next, the analysis software is applied to two weeks' worth of e-mail system logs as part of this thesis' experiment.

Chapter 4

Experiment and Results

Two weeks of system mail were analysed in the experiment. From over 32,000,000 syslog message entries 2,644,135 message views were reconstructed.

4.1. Overwhelmingly Delivered

The good news is that the majority of e-mail sent is delivered, and delivered from sources that have very low message rejections rates.

The first conclusion drawn from this is that most e-mail being sent from the system under test is not spam. This assertion is reinforced by the system's excellent reputation with major RBL providers.

Because the intent of this research is to identify customer accounts being used to spam, the rest of this chapter will focus on what is learned from the undelivered messages.

4.2. The Undelivered

Of the 2,644,135 message views extracted from the logs, 2,525,469 were successfully delivered. This leaves 115,666 messages or 4.37% undelivered in the two week period.

4.2.1. The Long Tail: less than 250 messages per week

Sources that did not send at least an average of 250 messages per week over the test period are removed from further analysis. These sources' volume is too low to draw any concrete conclusions.

The long tail of low volume users is removed from further consideration. This represents 94.2% of the sources sending mail, yet they account for 21.5% of the total volume moved,

and only 9.7% of the undelivered mail. Hence the conclusion that these low volume senders can be reasonably considered not to be spammers.

4.2.2. Smaller number of high volume sources

Senders attempting to deliver over 250 messages per week represent 779 of the 13379 unique sources seen, or 5.8%. In volume they are responsible for 78.5% of messages, and 90.3% or 104,495 of the undelivered mail.

This corpus exhibits a second long tail pattern. Of these senders, most exhibit an over 91% delivery rate. Plotting sources to percentage of undelivered mail demonstrates this effect in Figure 14.

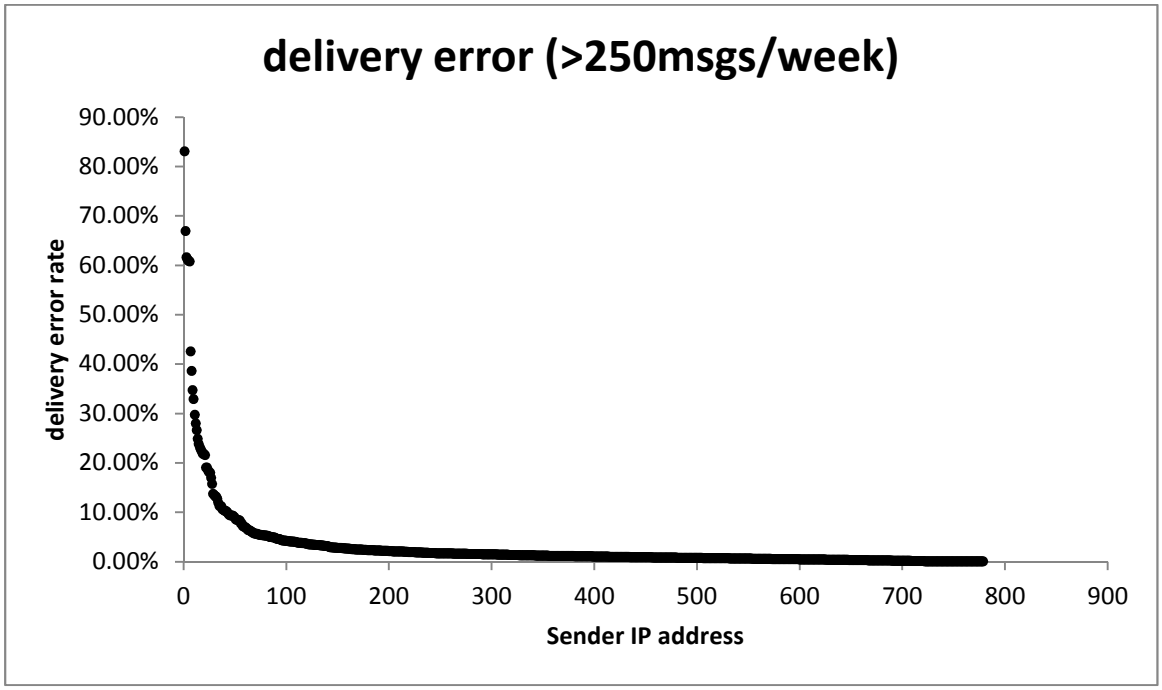

Figure 14: delivery error rate for sources over 250 messages/week

The undelivered data is more useful if expressed as delivery error rate to overall message volume, as demonstrated in Figure 15. This graph again shows that the majority of senders have a very low error rate and very high acceptance rate from foreign e-mail systems for their messages sent.

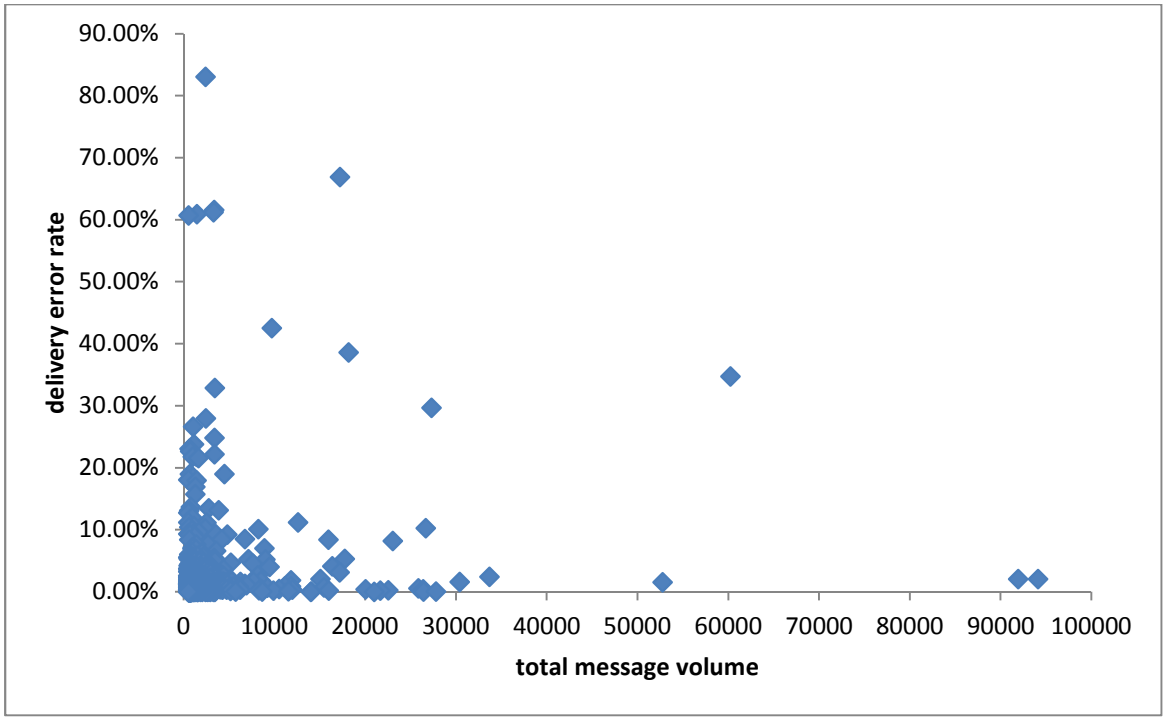

Figure 15: delivery error rate to total message volume

The output to focus on as spammer candidates are the high volume and high error rate sources, or the upper right quadrant of Figure 15.

4.2.3. Over 9% undelivered

To further reduce the number of sources under analysis, a second criterion is added to the data set: only senders for whom more than 9% of messages are undelivered will be scrutinized.

Again, this cut mostly removes senders with valid recipients from the mix.

Only 49 (0.37%) of 13379 total sources satisfy the condition of attempting to send more than 250 messages per week, while having over 9% undelivered.

Yet this small percentage of senders is responsible for 9% of the total volume of messages delivered in the study timeframe, and more importantly 53% of the volume of undelivered messages.

In depth analysis of this group was limited to the highest volume senders. For privacy reasons, of the whole 13,379 sender data set, only detailed message view information on the top 50 senders by total volume was kept.

One of this works' base hypotheses to test in a complex large scale e-mail platform is that a high volume of rejected messages translates to a very likely spammer sender. With the constraint that in-depth data was only kept for the top 50 overall senders, manual verification is only possible for the top 8 senders in the 9% undelivered sub-set.

Table 1: 8 sources with the highest ratio of undelivered messages

source	total	sent	bounce	defer	undelivered (B + D)	deliver %	undelivered %	HVS or LVS
1	60228	39330	17928	2970	20898	65.30%	34.70%	HVS
2	27322	19220	7250	852	8102	70.35%	29.65%	HVS
3	26662	23929	1224	1509	2733	89.75%	10.25%	HVS
4	18174	11162	6038	974	7012	61.42%	38.58%	LVS
5	17246	5709	8064	3473	11537	33.10%	66.90%	HVS
6	12618	11206	1298	114	1412	88.81%	11.19%	LVS
7	9718	5587	2208	1923	4131	57.49%	42.51%	LVS
8	8240	7408	791	41	832	89.90%	10.10%	unsure

For these 8 senders, the author is certain after manual analysis that the top 7 are spammers. Of these, four exhibit high volume spam patterns – a large volume of mail sent over a short time frame to a sequential list of recipients. Another three are low volume spammers, with messages trickled through over the entire duration of the experiment to a random list of recipients. The low volume senders also mutated their envelope source address on each send, the reasons for this cannot be precisely determined; however it would have the effect of hiding the spamming activity from analysis systems based solely on envelope sender addresses. The changed source address could also be used to track responses to the messages.

Only one sender cannot be clearly identified as a spammer, and could qualify as a poorly maintained mailing list. Looking at the detailed message data supports the poor mailing list hypothesis: the sender is a company that appears to only operate in one Australian State, and the majority of their recipients also appear to be in this same State.

4.2.4. A Viable Automated Cut-off Threshold

If an automated cut-off were to be devised from this data set for the e-mail system under study, it would be to block all sources that send over 4000 messages per week, and have over 25% rejected by the destination e-mail systems. This would amount to 5 out of 13379

sources, yet represent 5% of the total messages sent, and an astonishing 44.6% of all undelivered mail. These cut-off figures are designed to avoid false positives, and would not block uncertain cases such as the 8th sender in Table 1.

In other words, it is possible to find a cut-off level for spammer source detection in a complex e-mail system that is based on SMTP response codes alone. In this case with the over 4000 messages/week and over 25% rejected thresholds applied, almost half of the undelivered messages in the e-mail service providers outbound flows would be prevented from being sent.

For organizations that do not wish to implement automated blocking systems, the number of senders that meet this criteria (five, out of a customer base of over 100,000) is sufficiently low to validate manually.

4.3. Mailing Lists

Previous attempts to limit outbound spam have triggered on a source's message volume. The output of this work confirms that this is a poor metric to choose because it would block legitimate mailing lists while allowing a considerable volume of low volume spam to pass.

Two types of mailing list or mass mail out activity is found within the dataset used for this work. Either:

- One copy of a message is sent to the e-mail system, with a large set of envelope recipients – a "rcpt to:" entry for each. This method is most efficient in conserving client bandwidth, as the body of the message need only be sent once. However it also implies that each message must be the same, limiting its use to mailing lists
- The entire message is sent once per recipient. This creates a larger volume of individual messages moving through the e-mail system, and consumes more client bandwidth. This method allows the messages to be personalized for a mailing list, or for mass mail-outs such as those generated by monthly billing runs where thousands of customers would be sent an invoice.

.

High volume mailing list activity belongs in the bottom right quadrant of Figure 15: many messages are sent, but very few are rejected from the recipient mailsystems. The author hypothesises that vigorous anti-spam laws have made most legitimate companies and organizations place more stringent checks on their mail-outs, whereas spammers can perform

no such checks. The result remains: blocking by volume alone is not viable for the e-mail system operator as it will block considerable volumes of legitimate e-mail.

Chapter 5

Conclusion and Future Work

Identifying and reducing the volumes of outbound spam in high volume e-mail system is a worthwhile endeavour. The operators of the system benefit in improved reputation, as do the e-mail users who eventually receive less spam.

This work demonstrates that in commercial e-mail systems with a wide range of users, simple metrics such as volume alone is insufficient to determine whether a sender is a spammer. The dataset shows that there are ample senders distributing a high volume of e-mail in their legitimate business activities, or in mailing lists that would be trapped by volumetric analysis. Similarly, volumetric analysis would fail to detect low volume spammers.

Building a complete view of a sender's activities across all servers that are part of an outbound e-mail system is of value as it allows analysis of complex behaviours.

With this complete information it is possible to trace the SMTP reply codes issued from foreign servers back to individual senders. The results support the hypothesis that the existence of a large percentage of rejected messages in a sender's outbound e-mail flows is a good metric for identifying them as a spammer. The results further suggest that for every e-mail system, a cut-off value exists that could greatly reduce outbound spam with a low rate of false positives.

5.1. Future Work

It would be interesting to include further metrics when analysing outbound e-mail flows.

When selecting further metrics, the data source for these metrics must be taken into consideration. This work has relied on system logs, which restricts analysis to envelope data and very basic message information.

Message size is available in the system logs generated by Postfix and other MTAs. In analysing the data for this thesis, it was observed that message sizes in messages sent to a mailing list from non-spammer senders appeared to be identical or near identical. Slight variances could imply a minor change in the message for personalization, such as placing the recipient's name in the message body. However, some of the messages spammers sent to what appeared to be a mailing list varied greatly in size. The question is whether the variance is due to message padding – a countermeasure to content analysis – or if an entirely different payload is being sent. It may be that for a given user, messages sent within a short period of time with a large variance in individual message size is an indication of spammer activity. Validating this metric without the message content would be difficult, as it would be important to verify that sequential messages carry a similar body or intent.

Message content is not checked in the analysis done in this work. It would be interesting to run automated tests, such as applying the anti-spam or anti-virus software to outbound flows. This idea has been suggested in the past, and has often been rejected for being too computationally expensive. However the cost of processing power continues to decrease, which could make this viable. Or perhaps content analysis could be used to validate the results of other metrics, such as the recipient system SMTP response code used in this work. The computational cost may be acceptable if used in cases of uncertainty alone. For this work, uncertainty represents senders with over 2000 messages per week having over 10% rejection. Applying content scanning to this corpus would restrict this computationally expensive activity to 9% of the total outbound flow. The benefit would be the potential removal of up to 65% of outbound spam – a 20% increase.

Appendix A

Abbreviations

DNS	Domain Name Service
DNS MX record	Domain Name Service Mail eXchanger record
MTA	Mail Transport Agent. Software responsible for moving and queuing mail from one system or one process to another, usually over SMTP.
RBL	Realtime Block List
SMTP	Simple Mail Transport Protocol

Bibliography

[1] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage, “Spamalytics: An empirical analysis of spam marketing conversion,” in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 3–14.

[2] J. Klensin, “Simple Mail Transfer Protocol,” RFC 2821 (Proposed Standard), Internet Engineering Task Force, Apr. 2001, obsoleted by RFC 5321, updated by RFC 5336. [Online]. Available: http://www.ietf.org/rfc/rfc2821.txt

[3] J. Postel, “Simple Mail Transfer Protocol,” RFC 821 (Standard), Internet Engineering Task Force, Aug. 1982, obsoleted by RFC 2821. [Online]. Available: http://www.ietf.org/-rfc/rfc821.txt

[4] D. Crocker, “STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES,” RFC 822 (Standard), Internet Engineering Task Force, Aug. 1982, obsoleted by RFC 2822, updated by RFCs 1123, 2156, 1327, 1138, 1148. [Online]. Available: http://-www.ietf.org/rfc/rfc822.txt

[5] P. Resnick, “Internet Message Format,” RFC 2822 (Proposed Standard), Internet Engineering Task Force, Apr. 2001, obsoleted by RFC 5322, updated by RFCs 5335, 5336. [Online]. Available: http://www.ietf.org/rfc/rfc2822.txt

[6] “Windows Events,” Internet Engineering Task Force, Jul. 2009. [Online]. Available: http://msdn.microsoft.com/en-us/library/aa964766%28v=VS.85%29.aspx

[7] C. Lonvick, “The BSD Syslog Protocol,” RFC 3164 (Informational), Internet Engineering Task Force, Aug. 2001, obsoleted by RFC 5424. [Online]. Available: http://-www.ietf.org/rfc/rfc3164.txt

[8] R. Gerhards, “The Syslog Protocol,” RFC 5424 (Proposed Standard), Internet Engineering Task Force, Mar. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5424.txt

[9] D. Alperovitch, P. Judge, and S. Krasser, “Taxonomy of Email Reputation Systems,” in *ICDCSW ’07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*. Washington, DC, USA: IEEE Computer Society, 2007, p. 27.

[10] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, “Spamming botnets: signatures and characteristics,” in *SIGCOMM ’08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. New York, NY, USA: ACM, 2008, pp. 171–182.

[11] A. Ramachandran, D. Dagon, and N. Feamster, “Can DNS-based blacklists keep up with bots,” in *Conference on Email and Anti-Spam*, 2006.

[12] “MAPS, the Mail Abuse Prevention System.” [Online]. Available: http://www.mail-abuse.com/

[13] A. Ramachandran and N. Feamster, “Understanding the network-level behavior of spammers,” in *SIGCOMM ’06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2006, pp. 291–302.

[14] “The Spamhaus Project.” [Online]. Available: http://www.spamhaus.org/

[15] “Spamcop.net.” [Online]. Available: http://www.spamcop.net/

[16] “SURBL, the Spam URI Realtime Blocklists.” [Online]. Available: http://-www.surbl.org/

[17] K. Walsh and E. G. Sirer, “Fighting peer-to-peer SPAM and decoys with object reputation,” in *P2PECON ’05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*. New York, NY, USA: ACM, 2005, pp. 138–143.

[18] ——, “The case for an object-based peer-to-peer reputation system,” in *SOSP ’05: Proceedings of the twentieth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2005, pp. 1–2.

[19] W. N. Gansterer and M. Ilger, “Analyzing UCE/UBE traffic,” in *ICEC ’07: Proceedings of the ninth international conference on Electronic commerce*. New York, NY, USA: ACM, 2007, pp. 195–204.

[20] A. Pathak, Y. C. Hu, and Z. M. Mao, “Peeking into spammer behavior from a unique vantage point,” in *LEET’08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–9.

[21] R. Clayton, “Stopping spam by extrusion detection,” *To appear*, 2004.

[22] J. M. Pujol, R. Sangüesa, and J. Delgado, “Extracting reputation in multi agent systems by means of social network topology,” in *AAMAS ’02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2002, pp. 467–474.

[23] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, C.-W. Hu, and W. Hu, “A Behavior-based Approach To Securing Email Systems,” in *Mathematical Methods, Models and Architectures for Computer Networks Security*. Springer Verlag, 2003, pp. 57–81.

[24] S. J. Stolfo, S. Hershkop, C.-W. Hu, W.-J. Li, O. Nimeskern, and K. Wang, “Behavior-based modeling and its application to Email analysis,” *ACM Trans. Interet Technol.*, vol. 6, no. 2, pp. 187–221, 2006.

[25] S. Hershkop, “Behavior-based email analysis with application to spam detection,” Ph.D. dissertation, Columbia University, 2006.

[26] J. T. Goodman and R. Rounthwaite, “Stopping outgoing spam,” in *EC ’04: Proceedings of the 5th ACM conference on Electronic commerce*. New York, NY, USA: ACM, 2004, pp. 30–39.

[27] “Scribe,” Facebook Engineering, Oct. 2008. [Online]. Available: https://github.com/-facebook/scribe

[28] W. Venema, “Postfix,” IBM Research, Jan. 1998. [Online]. Available: http://-www.postfix.org/

[29] D. Luckham and B. Frasca, “Complex event processing in distributed systems,” *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*, 1998. [Online]. Available: http://reference.kfupm.edu.sa/content/c/o/-complex_event_processing_in_distributed__74987.pdf

[30] P. Dekkers, “Complex event processing,” Master’s thesis, Radboud University Nijmegen, 2007. [Online]. Available: http://dist.codehaus.org/esper/-CEP_MasterThesis_PaulDekkers_200709.pdf

[31] “ESPER Event Stream Intelligence.” [Online]. Available: http://esper.codehaus.org/