

Sheffield Hallam University
School of Computing & Management Sciences



Project and Professional Development

Final Year Project

Spam Detection and Filtering Techniques

Scott Clark

Group G (N&C)

April 2004

<http://www.scottclark.me.uk/>

mail at scottclark dot me dot uk

Abstract.....	4
Aims and Objectives.....	5
Problem Context.....	6
Methodology.....	9
Conventions.....	11
Glossary of Terms.....	12
Possible Solutions.....	18
Designated Sender Schemes.....	19
Sender Policy Framework (SPF).....	19
What is SPF?.....	19
How does SPF Work?.....	19
Advantages of SPF.....	20
Disadvantages of SPF.....	20
Who is using SPF?.....	21
Evaluation of SPF.....	21
Caller-ID for Email.....	22
What is Caller-ID for Email?.....	22
How does Caller-ID for Email work?.....	22
Advantages of Caller-ID for Email.....	25
Disadvantages of Caller-ID for Email.....	25
Who is using Caller-ID for Email?.....	25
Evaluation of Caller-ID for Email.....	25
Evaluation of Designated Sender Schemes.....	26
DNS Blacklists.....	27
What are DNS Blacklists?.....	27
Advantages of DNS Blacklists.....	28
Disadvantages of DNS Blacklists.....	28
Evaluation of DNS Blacklists.....	29
Bayesian Filtering.....	30
What is Bayesian Filtering.....	30
Advantages of Bayesian Filtering.....	31
Disadvantages of Bayesian Filtering.....	31
Evaluation of Bayesian Filtering.....	31
Spam Blackholes.....	32
What are Spam Blackholes.....	32
Advantages of Spam Blackholes.....	33
Disadvantages of Spam Blackholes.....	33
Evaluation of Spam Blackholes.....	33
Spam Filtering in Action.....	34
Software Review.....	34
Mail Washer.....	35
Experiment Notes.....	36
McAfee Spam Killer.....	37
Experiment Notes.....	38
Norton Anti-Spam.....	39
Experiment Notes.....	39
Spam Assassin.....	40
Experiment Notes.....	42
Cloudmark SpamNet.....	43

Experiment Notes	44
Spam Bayes.....	45
Experiment Notes	46
Test Results.....	47
Mail Washer.....	47
Test 1.....	47
Test 2.....	47
Test 3.....	48
Test 4.....	48
McAfee Spam Killer.....	49
Norton Anti-Spam.....	49
Spam Assassin	50
Test 1.....	50
Test 2.....	50
Cloudmark SpamNet.....	51
Spam Bayes.....	51
Total Missed	52
Price Comparison.....	53
Discussion of Results.....	54
Evaluation of Testing.....	56
Project Recommendations	57
Evaluation of Recommendations	59
Practical Implications and Further Development	60
Client Contact and Evaluation	63
Progress against Project Plan.....	64
Critical Discussion of Own Work.....	67
References.....	69
Bibliography	71
Appendix 1 – Project Specification	73
Appendix 2 – The 1 st Spam Message Sent	76
Appendix 3 – SPF Syntax	77
Appendix 4 – VMWare Images / Spam Messages	78
Appendix 5 – Spam Filtering Script	79
Appendix 6 – Spam Questionnaire	80

Abstract

How many times have you opened your email and been inundated with emails advertising penis enlargement, Viagra and Weight loss pills?

The problem of unsolicited commercial email (UCE) or more commonly referred to as spam, is growing.

In August 2003, the company Message Labs, who are a provider of email security services to businesses, intercepted more than 91.2 million spam emails. They say that this is a global spam ratio of 1 in every 2.8 emails.

August 2003 also saw a massive amount of email borne viruses. The outbreak of Sobig.F was reported, by the BBC and others, to be one of the fastest growing viruses ever. Message Labs reported that in the first 24 hours of the virus outbreak, more than one million copies of the virus were detected.

The need for comprehensive scanning of emails is extremely important. This project aims to compare and contrast the different ways of stopping the inconvenience of spam emails. This will involve looking at existing systems, both on the SMTP server and on the desktop.

This project will investigate different methods to detect and prevent spam.

Different 'off the shelf' packages will also be tested to compare their effectiveness in detecting spam messages and preventing them reaching your inbox.

Aims and Objectives

The student is required to:

1. Explore and understand the different methods and tools available to undertake this project.
2. Explore how email addresses are harvested by spam senders.
3. Research different methods of filtering / scanning email both on the server and desktop.
4. Using a methodical approach, test the different systems and compare features etc.
5. Research different methods of trying to prevent unsolicited emails
6. Produce a set of recommended methods to prevent such threats.
7. Critically evaluate the recommendations produced and evaluate their usefulness.
8. Critically evaluate the usefulness of the methods and tools used to create the recommendations and illustrate any drawbacks found
9. Suggest future enhancements and development of the recommendations produced
10. Produce a report of the project.

The deliverable for this project will be a report outlining a set of recommendations to help prevent spam and virus attacks.

Problem Context

Spam is the term now generally used to refer to unsolicited electronic messages, usually transmitted to a large number of recipients. They usually, but not necessarily, have a commercial focus, promoting or selling products or services; and they share one or more of the following characteristics:

- They are sent in an untargeted and indiscriminate manner, often by automated means;
- They include or promote illegal or offensive content;
- Their purpose is fraudulent or otherwise deceptive;
- They collect or use personal information in breach of several countries privacy laws;
- They are sent in a manner that disguises the originator;
- They do not offer a valid and functional address to which recipients may send messages opting out of receiving further unsolicited messages.

Not all bulk email is spam. Bulk email would probably not be generally regarded as spam if it:

- Is sent to recipients who have previously dealt voluntarily with the sender before and, on the basis of that existing relationship, can reasonably be assumed by the sender to be prepared to accept messages of the type being sent;
- Does not promote or include illegal content;
- Is not deceptive in any way that breaches common law or statute law;
- Does not collect or use personal information in breach of privacy laws.

While spam has increased in prominence in recent years, growing from a minor nuisance to a significant problem, its existence actually predates the Internet. It has been the subject of discussion since at least 1975, when Jon Postel wrote in RFC 706 about the junk mail problem. One of the first recorded instances of spam dates back to 1st May 1978, when Digital Equipment Corporation (DEC) spammed ARPANet users about new DEC products. The text of this spam message is in Appendix 2.

Probably the first major commercial spamming occurred in 1994, when two lawyers posted a message advertising their services to several thousand newsgroups on UseNet, the world's largest online conferencing system. Then, as now, the reaction to spam was overwhelmingly negative, although it was seen as an occasional nuisance and did not pose any real threat. There were even then, though, some instances where spamming was used to maliciously interrupt services by overloading email servers.

In 2003, Paul Wood, the Chief Information Security Analyst at Message Labs writes about the damage that spam does to profitability, important in the business community. He writes:

The first casualty of the spam epidemic is employee productivity. Staff who have to wade through their email inboxes to differentiate the relevant email from the rubbish are at an immediate time-management disadvantage. And who knows how much more time they might be persuaded to waste if they were drawn by a particular spam offer? They would lose concentration and be distracted from their work, resulting in a considerable break in productivity.

A Gartner survey in 2002 revealed that staff were already spending an average 49 minutes a day (that's 10 per cent of their time) just managing their email – and that 34 per cent of business email being received was spam. Given the massive escalation of spam since then, it may be assumed that email management has become an even more time-consuming activity at work.

It has to be paid for; and selection of bandwidth from your ISP is a business decision tailored to your anticipated business usage. If just a quarter of your incoming mail is unsolicited spam, the unexpected volume will already be throttling your Internet connection. That slow-up in email delivery can soon start to impact on your hard-won competitive edge. You can increase bandwidth to maintain speed, of course – but do you really want to pay more, just so that spammers can get faster access to your people?

Volumes of pornographic spam are increasing. So is the pressure on employers to protect their people from exposure to offensive or disturbing email-borne material. There have already been well publicised instances where employees have successfully brought legal cases against their employers, principally in the area of sexual harassment stemming from the abuse of email. However, given the continuing growth in pornographic spam, the hazard is already beginning to extend worryingly into the realms of unsolicited email.

European Commission figures for 2001 estimate that the cost of spam globally is around £6.4bn a year in connection charges alone. With in excess of 544 million Internet subscribers worldwide, that's an average of nearly £12 per user per year.

The threat to ISPs (Internet Service Providers), businesses and consumers is particularly severe in the US. One large ISP reported receiving nearly two million spams each day from a promotions company until an injunction was obtained to prevent it. If you assume that each user spends only ten seconds identifying and deleting spam messages, the amount of connection time being squandered through that single ISP was in the order of 5,000 hours every day.

As part of a six-month investigation of spam by the Centre for Democracy and Technology, a US not-for profit Internet privacy group, 250 email addresses were set up. These attracted 10,000 emails, of which 8,842 were unsolicited.

Potential cost of spam

<i>Number of Users</i>	<i>Lost Productivity (minutes per month)</i>	<i>Lost Productivity (per month)</i>
<i>25</i>	<i>492</i>	<i>£164</i>
<i>100</i>	<i>1,966</i>	<i>£655</i>
<i>250</i>	<i>4,916</i>	<i>£1,639</i>
<i>500</i>	<i>9,831</i>	<i>£3,277</i>
<i>1000</i>	<i>19,663</i>	<i>£6,554</i>
<i>2000</i>	<i>39,325</i>	<i>£13,108</i>
<i>5000</i>	<i>98,313</i>	<i>£32,771</i>

N.B. Using the March 03 figures for spam of 36.3%, for a company with the following numbers of employees (hourly cost approx. £20), each receiving on average 30 mails per day, taking an average of 5 seconds to read and delete spam emails.

As you can see the problem of spam affects businesses, it costs them both time and money. Effective Anti-Spam measures are needed to address the problem before an employee spends more time sorting spam than they actually do working.

Methodology

In the later part of this report, different tools are evaluated to find out their effectiveness.

The applications were tested under the following conditions:

A clean install of Windows 2000 Professional was created under VMWare into which the Windows based applications were installed.

Another VMWare image was created running Debian GNU/Linux 3.0r2 (Woody), this was to both test the Linux based applications and to run as a POP3 server from which the various client programs will download messages from.

Both images are included on the DVD-R in Appendix 4.

The reason that virtual images were created was to ensure the integrity of the systems, to ensure that any possible viral infections would be confined to that image. The Association of Computing Machinery (ACM) and British Computer Society (BCS) Code of Conduct both say that a professional should avoid harm to others; if a viral infection or spam messages were relayed through a test machine then this could harm others. Using a virtual environment ensured that the machines did not have access to the outside world and were unable to contribute to the spam problem.

To test the effectiveness of the applications, both spam and normal messages were needed. The normal messages are a selection of mail received over the past 3 years, the spam messages are a selection from <http://www.spamarchive.org> and some that my mail server has classified as spam. The selection of messages is included on the DVD-R in Appendix 4.

The breakdown of messages is as such:

10 batches; each of 150 emails, 1500 in total.

The following table shows the breakdown of spam and clean messages in each batch.

Batch	Spam	Clean
0	100	50
1	105	45
2	95	55
3	120	30
4	112	38
5	77	73
6	92	58
7	133	17
8	140	10
9	70	80

Totals 1044 456 1500

These figures were chosen to represent the typical amount of spam that was received on a regular basis.

The messages were broken down into batches to test the effectiveness of software to learn about the type of messages received and to detect similar messages.

During the tests, the email was copied onto the POP server, and then the client software was used to access and download the files onto the client. What settings were used for each test will be discussed later in the report.

Conventions

Text in boxes indicates the input of a command and its output.

Text in *italics* indicates the command typed into the command line.

Glossary of Terms

ARPAnet – Advanced Research Projects Agency Network

ARPAnet was the precursor to the Internet.

It was developed in the late 60's and early 70's by the US Department of Defence.

It was an experiment in wide-area-networking to connect together computers that were each running different systems so that people at one location could use computing resources at another.

Bounce –

A bounce is a message that is generated by a MTA to notify the sender that the user to whom they sent email does not exist. A bounce does not have an envelope sender.

CIDR – Classless Inter-Domain Routing

CIDR is a system of allocating IP addresses more effectively than the old allocations of Class A, B or C. It consists of an IP address followed by a prefix, i.e. 82.68.38.136/29; this information encapsulates the following range of IP addresses, 82.68.38.136 – 82.68.38.142.

Further information about CIDR is beyond the scope of this project so will not be discussed here.

DNS – Domain Name System

The Domain Name System is the system that translates Internet domain names into IP Addresses. A "DNS Server" is a server that performs this kind of translation.

It acts like a telephone directory.

Which is easier to remember, 212.13.208.215 or www.scottclark.me.uk? DNS takes the name given, and returns the numerical IP address.

DNS Zone –

A DNS zone is a text file containing details of a domain, who administers it, the name servers used for DNS resolution, the hosts that handle mail for the domain and lists the IP addresses of any hosts in that domain. This is an example of a DNS zone, showing the SOA, NS records, MX records, a TXT Sender Permitted From record and some A records.

```
$TTL      3600
@         IN      SOA     ns1.thunderweb.co.uk. hostmaster.thunderweb.co.uk. (
                                2004042369 ; Serial
                                3600      ; Refresh
                                1800      ; Retry
                                604800    ; Expire
                                3600)     ; Negative Cache TTL
;
@         IN      NS      ns1.thunderweb.co.uk.
@         IN      NS      ns2.thunderweb.co.uk.
@         IN      MX      10   mx1.thunderweb.co.uk.
@         IN      MX      20   mx2.thunderweb.co.uk.
@         IN      TXT     "v=spf1 a -all"
@         IN      A       212.13.208.215
www      IN      A       212.13.208.215
```

An SOA Record is the ‘Start of Authority’, it gives details of who administers the domain and details about with version of the zone that this is.

NS records specify which DNS servers hold details of this domain.

MX (Mail eXchanger) records specify which servers accept email for this domain.

TXT records allow you to add detail about domains; in this case it is a SPF record. SPF will be discussed later in this report.

A records exist to convert names into IP addresses.

There are also other types of DNS records, but they are beyond the scope of this project so will not be discussed here.

Envelope Sender –

An envelope sender is the addresses used during the MAIL FROM: part of a SMTP conversation. It can be different from the From: line in the message headers. It can easily be forged, which leads to junk email addressed from innocent people.

IETF – Internet Engineering Task Force

The Internet Engineering Task Force is a loosely self-organized group of people who contribute to the engineering and evolution of Internet technologies. It is the principal body engaged in the development of new Internet standard specifications / protocols.

IMAP – Internet Message Access Protocol

IMAP is gradually replacing POP3 as the main protocol used by mail clients in communicating with mail servers.

Using IMAP, a MUA can not only retrieve email but can also manipulate message stored on the server, without having to actually retrieve the messages. So messages can be deleted, have their status changed, multiple mail boxes can be managed, etc.

IP Address –

An IP address is an identifier for a computer / device on a network. Except in certain cases, an IP address is globally unique, no other computer / device can have the same IP address. With IPv4 an IP address is in the form 82.68.38.138, the decimal representation of a 32-bit number.

IPv4 – Internet Protocol, version 4

The most widely used version of the Internet Protocol (the "IP" part of *TCP/IP*.)

IPv4 allows for a theoretical maximum of approximately four billion IP Addresses (technically 2^{32}), but the actual number is far less due to inefficiencies in the way blocks of numbers are handled by networks. The gradual adoption of IPv6 will solve this problem.

IPv6 – Internet Protocol, version 6

IPv6 is the successor to IPv4. Already deployed in some cases and gradually spreading, IPv6 provides a huge number of available IP Addresses - over a sextillion addresses (theoretically 2^{128}).

IPv6 allows every device on the planet to have its own IP Address.

As IPv6 is beyond the scope of this project, a further definition will not be included.

ISP – Internet Service Provider

An ISP is a company that provides access to the internet in some form. It connects your computer / network into there network, providing you with connectivity to the rest of the Internet.

Joe Jobbing –

A Joe Job is essentially spam designed to look like it's coming from someone else.

In 1996, the website Joe's Cyberpost (joes.com), a hosting company, had a user who spammed newsgroups advertising his page.

His account was killed for a breach of joes.com's Acceptable Use Policy (AUP).

The user then went on to send millions of spams; but this time he forged the return address to make it look like Joe Doll, the owner of joes.com, was doing the spamming.

Well, joes.com was inundated with complaints from newsgroup users and email account holders. They were also Denial of Service (DOS) attacked by a bunch of angry hackers too lazy to properly check the source of the spam.

Joes.com's server died on or about New Year's Day, 1997.

Since then, the term Joe Job has referred to anyone trying to pull the same trick.

MTA – Message Transfer Agent

A MTA is the program responsible for receiving incoming messages and sending outgoing messages. The MTA transfers messages between computers normally using SMTP. They are hidden from the average computer user. The MTA is more commonly referred to as the mail server program. Examples of MTA's include qmail and sendmail on Unix / Linux machines and Microsoft Exchange on Windows machines.

MUA – Mail User Agent

A MUA is a program that an end user uses in order to send and receive email. They communicate with MTA's to send and receive email. Examples of MUA's include Microsoft Outlook and Mozilla Thunderbird

Open Relay

An open relay is an SMTP server that allows anybody to send messages through it, regardless of if they are authorised or not. Open relays allow for spammers to send email and make it look like it's from another company.

POP3 – Post Office Protocol (version 3)

POP3 is a protocol for a MUA to communicate with a mail server in order to download messages.

RFC – Request for Comments

A RF is the name of the result and the process for creating a standard on the Internet. New standards are proposed and published on the Internet, as a Request for Comments. The proposal is reviewed by the IETF, a consensus-building body that facilitates discussion, and eventually a new standard is established, but the reference number/name for the standard retains the acronym RFC, e.g. the official standard for e-mail message formats is RFC 822.

SMTP – Simple Mail Transfer Protocol

SMTP is the main protocol used for MTA to communicate with one another and for MUA's to send mail to a MTA. It is defined in RFC 821 and modified by numerous later RFC's.

SMTP is based on a series of commands that send a message from other server to another. A typical SMTP conversation is shown below: (Text in **bold** is the response from the server, *italics* is what the sender sends.)

```
telnet topcat.thunderweb.co.uk smtp
220 topcat.thunderweb.co.uk ESMTP
helo yogi.scottclark.me.uk
250 topcat.thunderweb.co.uk
mail from: test_message@scottclark.me.uk
250 ok
rcpt to: mail@scottclark.me.uk
250 ok
data
354 go ahead
To: mail@scottclark.me.uk
From: test_message@scottclark.me.uk
Subject: Test Message

Hello :)

.
250 ok 1082883311 qp 416
quit
221 - so long and thanks for all the fish... - topcat.thunderweb.co.uk
```

Virus –

A virus is a piece of computer programming code that makes copies of itself without any conscious human intervention.

Some viruses do more than simply replicate themselves, they might display messages, install other software or files, delete software or files, etc.

VMWare –

VMWare is an application that allows you to run virtual machines on top of your existing software. It is used in this project to provide a Windows 2000 and a Linux machine running at the same time as Windows XP on the same physical machine. Further information is available from the vendor, at <http://www.vmware.com>.

XML – eXtensible Markup Language

A widely used system for defining data formats. XML provides a very rich system to define complex documents and data structures such as invoices, molecular data, news feeds, glossaries, inventory descriptions, real estate properties, etc.

As long as a programmer has the XML definition for a collection of data (often called a "schema") then they can create a program to reliably process any data formatted according to those rules.

Possible Solutions

There are quite a few different solutions that are either in use or have been recently designed in order to reduce / eliminate the problem of spam. This section of the report looks into several different mechanisms, how they work, their advantages and disadvantages and evaluates how they could be useful.

Designated Sender Schemes

Sender Policy Framework (SPF)

What is SPF?

Sender Policy Framework is an example of a designated sender scheme, whereas a domain owner can publish records that specify the name and addresses of servers that are allowed to send email from that domain. It is being created by a group of people around the world, it is an open standard, and no licence is required to use it.

The Acronym SPF originally stood for Sender Permitted From, which better explains what it does, in February 2004 it was changed to Sender Policy Framework.

SPF is primarily an anti-forgery effort. Any benefits in the area of reduced spam and viruses are pleasant side-effects. If SPF causes spammers to send mail from their own domains, it will be easier to identify and block those domains.

How does SPF Work?

The owner of a domain publishes a DNS record that lists the servers that can send email from a domain. This record is simply a TXT record in the DNS zone. As an example, I'll look at amazon.com.

To test for a SPF record, you need to query the DNS system for a TXT record.

```
dig amazon.com any | grep TXT
amazon.com.      7035  IN    TXT   "v=spf1 ip4:207.171.160.0/19 ?all"
```

In this example the administrator of amazon.com has specified that any IP that falls within the 207.171.160.0/19 CIDR range, which is 207.171.160.0 - 207.171.191.255, is allowed to send email with an envelope sender of address@amazon.com.

When an incoming server receives an email, it can query the SPF record and determine whether or not the email was sent from an amazon.com IP address or not. If it was, then the server assumes that the email is genuine and lets it through. What it does if the email was not sent from an authorised server is down to the individual implementation of SPF or the server administrator.

In a spam scoring system, where a piece of software gives points to certain features of an email, the system will give a SPF failing email a higher score.

SPF can tell the scanning software one of 4 things.

1. The sender is good; they have announced that they send mail from that IP.
2. The sender is bad, the purported sender has announced a list of IP's but this email isn't from one of them.
3. The sender may be good or bad, the sender domain is currently testing there SPF record.
4. SPF doesn't know, the sender domain has not published any SPF records.

The syntax of a SPF record is given in Appendix 3.

Advantages of SPF

One major advantage of SPF is that it allows a system administrator to specify which IP addresses are allowed to send email from a particular domain, this helps detect unauthorised people and viruses using your domain for there mail.

As SPF protects the email return path, users protected using Sender Policy Framework will stop getting bounce messages for messages they didn't send, for example, "Your computer sent us a virus".

Another advantage of SPF is that it is an open project; anybody can comment and contribute to the system. It is currently on its way to becoming a RFC as it has been submitted to the IETF.

Disadvantages of SPF

The major disadvantage regarding Sender Policy Framework is that it breaks forwarding. Most MTA's offer the facility to forward email from one address to another. For example when you are on holiday you may set up your company mail server to forward all emails to a web based system so that you can access them on the road.

When the email is forwarded, the envelope sender is preserved and sent to its new destination. If the receiving MTA checks for a SPF record for the sender, they will probably see that the IP address of your company server is not permitted to send email from that domain and mark it as a forgery.

This can be fixed by remailing instead of forwarding. Remailing is where the message is kept in tact but the envelope sender is changed to one containing the outgoing server.

Also, what's to stop a spammer buying a throwaway domain and publishing SPF records that say which IP's the spammer uses, or saying the entire internet can send mail.

Who is using SPF?

According to the SPF Registry, as of the end of March 2004, over 10,000 domains had published SPF records, including big companies and organisations, such as amazon.com, gnu.org, google.com, oreilly.com, perl.org, oxford.ac.uk, symantec.com and w3.org.

One of the biggest ISP's in the world, America Online (AOL) is also publishing an SPF record.

Evaluation of SPF

I personally think that SPF is a good idea in principle; it allows a domain owner to publish records stating what machines can send email from your domain; this will help to prevent virus notifications and Joe-Jobbing. Once more MTA's and MUA's understand how to check for SPF records, the amount of junk email that one receives should be reduced.

As discussed previously, a spammer could publish SPF records for there spam domain, with this in mind, SPF is part of the overall anti-spam solution, it will complement other technologies in the bid to reduce the amount of spam received.

Caller-ID for Email

What is Caller-ID for Email?

Caller-ID for Email is another designated sender scheme, whereas a domain owner can publish records that specify the name and addresses of servers that are allowed to send email from that domain. It was created by Microsoft, who have licensed the system for people to use.

Like SPF, Caller-ID for Email is primarily an anti-forgery system; whereas a domain administrator can list the ranges of IP addresses that can send email from that domain.

How does Caller-ID for Email work?

The owner of a domain publishes a DNS record that lists the servers that can send email from a domain. This record is simply a TXT record in the DNS zone.

To test for an email policy, you need to look up the TXT record for `_ep.domainname`.

As an example, I'll look at `amazon.com`.

```
dig _ep.amazon.com any / grep TXT
_ep.amazon.com.      6450 IN    TXT      "<ep xmlns='http://ms.net/1' testing='true'>
<out><m><r>207.171.160.0/19</r></m></out></ep>"
```

What is produced is an XML document;

```
<ep xmlns='http://ms.net/1' testing='true'>
<out>
  <m>
    <r>207.171.160.0/19</r>
  </m>
</out>
</ep>
```

Again, this lists the range of IP addresses that can send email with an envelope sender of `amazon.com`.

To look at a more complex example of Caller-ID for Email, I'll look at Microsoft's Hotmail service. Again, to start looking for the Email Policy, you query the DNS system for a TXT record for `_ep.hotmail.com`.

dig _ep.hotmail.com txt

```
<ep xmlns='http://ms.net/1' testing='true'>  
<out>  
  <m>  
    <indirect>list1._ep.hotmail.com</indirect>  
    <indirect>list2._ep.hotmail.com</indirect>  
    <indirect>list3._ep.hotmail.com</indirect>  
  </m>  
</out>  
</ep>
```

No actual IP addresses in this at the moment, but it says that there are three lists that should be queried to get the Email Policy.

This time, to get the email policy, you take the indirect name from the previous query and search for a TXT record for `_ep.list1._ep.hotmail.com`.

```
dig _ep.list1._ep.hotmail.com txt
<ep xmlns='http://ms.net/1' testing='true'>
<out>
<m>
<r>209.240.192.0/19</r>
<r>65.52.0.0/14</r>
<r>131.107.0.0/16</r>
<r>157.54.0.0/15</r>
<r>157.56.0.0/14</r>
<r>157.60.0.0/16</r>
<r>167.220.0.0/16</r>
<r>204.79.135.0/24</r>
<r>204.79.188.0/24</r>
<r>204.79.252.0/24</r>
<r>207.46.0.0/16</r>
<r>199.2.137.0/24</r>
<r>199.103.90.0/23</r>
</m>
</out>
</ep>
```

This policy document lists the CIDR ranges of IP addresses that can send email with a sender of hotmail.com.

This is repeated for the other two lists of IP addresses to give the entire range of addresses for hotmail.com.

Advantages of Caller-ID for Email

Like SPF, the major advantage of Caller-ID for email is that it allows a system administrator to specify which IP addresses are allowed to send email from a particular domain, this helps detect unauthorised people and viruses using your domain for there mail.

It has been developed by Microsoft, which will mean that Microsoft Products will start to support the technology; this will probably lead to an increase in use.

Disadvantages of Caller-ID for Email

Being developed by an in-house team at Microsoft; this system has not had the same amount of discussion / critique that SPF has had. It is likely that the standard will be driven by Microsoft's future needs and desires. It is covered by a patent licence, where patent issues may stall the implementation.

Caller-ID for Email has not been submitted to the IETF.

Caller-ID for Email uses XML for the list of allowed addresses, this makes it much bulkier than other designated sender schemes such as SPF. The DNS system works more efficiently when the data packets are small.

Who is using Caller-ID for Email?

It seems that not many people have implemented Caller-ID for Email on there mail systems. Microsoft is not keeping a list of people who have introduced the records; the only domains that I know of to use Caller-ID for Email are amazon.com, microsoft.com and amazon.com.

Evaluation of Caller-ID for Email

Caller-ID for Email is also a good idea in principle, as it allows a domain owner to publish records about there outgoing mail servers.

The problem with it is that it is not under open discussion, Microsoft have created the system which will either impede or push its uptake. At the moment, the only other company that can be found publishing records apart from Microsoft itself is Amazon. There needs to be more uptake before it can be used properly.

Evaluation of Designated Sender Schemes

At the moment there are 2 competing Designated Sender Schemes; Microsoft's Caller-ID for Email and Sender Policy Framework.

At the moment, SPF has more of an uptake, with over 10,000 domains publishing records. I can only find reference to 3 domains having Caller-ID records, 2 of them being Microsoft's own sites.

Having a look at the Caller-ID for Email records for hotmail.com reveal over 1 million IP addresses listed that can send email from hotmail.com, that's a lot of IP's, surely there can't be that many machines sending email from Hotmail?

The idea being Designated Sender Schemes is a good one, allowing a domain owner to publish records about which mail servers can send email from that domain.

There main goal is to prevent email forgery, as most spam messages have forged senders it will allow spam to be identified easily.

As stated previously, SPF or Caller-ID for Email are not the entire solution to preventing spam, they are part of the overall plan. Once they have a high uptake, and with the addition of other technologies, the amount of spam received in your inbox should drop. Only the future will tell.

DNS Blacklists

What are DNS Blacklists?

DNS Blacklists are list of IP addresses of machines that have a reputation for producing lots of spam. There are currently lots of lists for different purposes. There are ones that list ranges of dialup IP's, ones that list IP's of open relays and ones that send lots of spam. The way they work is that the receive SMTP server checks the IP and/or domain name of the sending machine in a blacklist.

In this example, I will use the SpamHaus SBL blacklist. (SpamHaus 2004)

The following 2 IP addresses will be tested, 81.132.113.250, a BT Openworld IP which was added to the SpamHaus blacklist on April 6th 2004 and 212.13.208.215, which is the IP address of my mail server.

To test for an entry in the RBL the server sends a DNS request for the IP address backwards.

```
nslookup 250.113.132.81.sbl.spamhaus.org
```

Non-authoritative answer:

Name: 250.113.132.81.sbl.spamhaus.org

Address: 127.0.0.2

```
nslookup 215.208.13.212.sbl.spamhaus.org
```

** server can't find 215.208.13.212.sbl.spamhaus.org: NXDOMAIN

In the 1st example an IP address is returned, meaning that the queried IP is on the RBL, the 2nd example a NXDOMAIN was returned, meaning that the queried IP was not on the RBL.

Depending on which blacklist is used, the IP address that is returned to signify a known spammer can be different depending on why that address was listed. For example SpamHaus returns 127.0.0.2 if the IP is a known spammer, whilst it returns 127.0.0.4 for machines known to be infected with a Trojan.

If the IP is listed in the RBL, a TXT record can normally be found in the same zone giving a reason for listing this IP.

```
dig 250.113.132.81.sbl.spamhaus.org txt
```

```
250.113.132.81.sbl.spamhaus.org. IN TXT "http://www.spamhaus.org/SBL/sbl.lasso?query=SBL13938"
```

In this example, the reason links to a page which includes a copy of the message that led to the inclusion of that IP.

DNS Blacklists can be used to block mail entirely from the addresses listed by denying connection attempts from those IP addresses. They are more commonly used in spam scoring systems, such as Spam Assassin, where additional points are given to a message if the IP address is listed in a blacklist.

Advantages of DNS Blacklists

One of the main advantages of using a DNS Blacklist is that it can block a substantial amount of mail. I've been using the SpamHaus RBL list since the start of March 2004, and as of April 2004, my mail server had blocked over 5700 mails! That's in just one month! So not accepting these mails leads to less traffic being sent over your network connection, if you pay for bandwidth by the GB, you will see less traffic thus saving money.

Disadvantages of DNS Blacklists

The main disadvantage of DNS blacklists is that a blacklist provider can block an entire net block range instead of just an individual IP. For example, a user on the Blueyonder ISP may send out spam from their machine, a RBL provider may block the entire net block meaning that servers that use that RBL will block legitimate emails from this entire range of IP's and not just the spam.

There have been many occasions in the past where an ISP's mail server has been blacklisted, and all mail that is relayed through it has either been blocked or additional points added in the scoring process.

Evaluation of DNS Blacklists

DNS Blacklists are one part of the overall spam solution; they allow you to identify likely spam candidates from the network / IP address that the mail was delivered from. As the statistics from my mail server show, the amount of messages blocked in just one month is very high, and that was just for a small mail server with 10 domains, imagine how much spam could be blocked by aol.com, hotmail.com and the likes.

Microsoft (2004) says that as of *'mid-2003, about 83% of the email messages received by Microsoft Hotmail on a typical day are spam'* *'That's around 2.5 billion out of nearly 3 billion messages'*

That's a lot of messages, and if these were blocked via a DNS Blacklist then the amount of spam that ended up in your inbox would be greatly reduced!

Using a DNS Blacklist had reduced that number of messages that my server receives, so that has to be a good thing!

Bayesian Filtering

What is Bayesian Filtering

Bayesian spam filters calculate the probability of a message being spam based on the contents of it. Bayesian filters learn from both good and spam messages, resulting in an adapting and efficient anti-spam approach.

The way that these systems work is that you feed into it copies of your email, both good and bad, marking saying whether they are either good or bad. The system takes the characteristics of these mails then analyses your incoming email.

Tschabitscher writes:

“Think about how you detect spam. A quick glance is often enough. You know what spam looks like, and you know what good mail looks like. The probability of spam looking like good mail is around... zero.

Wouldn't it be great if automatic spam filters worked like that, too?”

That's exactly how Bayesian filtering works, once trained it knows what good email looks like and most importantly it knows what spam messages look like.

The characteristics a Bayesian spam filter can look at can be

- the words in the body of the message
- its headers (senders and message paths, for example)
- other aspects such as HTML code (like colours)
- word pairs, phrases
- meta information (where a particular phrase appears, for example).

If a word, "Cartesian" for example, never appears in spam but often in your legitimate mail, the probability of "Cartesian" indicating spam is near zero. "Toner", on the other hand, appears almost exclusively in spam. "Toner" has a very high probability of being found in spam, not much below 1 (100%).

When a new message arrives, it is analyzed by the Bayesian spam filter, and the probability of the complete message being spam is calculated using the individual characteristics.

Let's say a message contains both "Cartesian" and "toner". From these words alone it's not yet clear whether we have spam or a legitimate message. But other characteristics will (most probably) indicate a probability that allows the filter to classify the message as either spam or good mail.

The characteristics of legitimate mail are just as important for the Bayesian spam filtering process as the spam is. If the filters are trained specifically for every user, spammers will have an even harder time working around most people's spam filters, and the filters can adapt to almost everything spammers try.

Spammers will only make it past well-trained Bayesian filters if they make their spam messages look perfectly like the ordinary email everybody may get. Spammers do not usually send such ordinary emails, I presume, because they don't work. So chances are they won't be doing it when ordinary, boring emails are the only way to make it past the anti-spam filters.

An example of Bayesian Filtering will be used in the practical part of this project.

Advantages of Bayesian Filtering

The major advantage of Bayesian Filtering is that it learns about the type of messages that you receive, making it more effective against detecting spam for the longer you use it. In a trained system there are very few false positives, where a legitimate mail is wrongly marked as spam, or a spam message being marked as legitimate.

Disadvantages of Bayesian Filtering

The only real disadvantage of a Bayesian Filter is that they need to be trained properly in order for them to work at their most effective. This training can take time so they are not effective straight out of the box.

Evaluation of Bayesian Filtering

Once your Bayesian Filter has been trained, it will be able to detect the majority of spam that you receive. A Bayesian Filter will adapt automatically to the mail passed through it making it a good choice for inclusion in a MUA.

Again, they are one part of the overall solution to reducing the spam problem, they are effective in their job, and using other technologies the amount of spam you receive can be reduced.

Spam Blackholes

What are Spam Blackholes

Spam Blackholes work hand in hand with DNS Blacklists.

The way a spam blackhole works is someone posts messages on websites, Usenet, forums etc, showing their email address. The email address they use is generally a machine account that detects who sent the spam and passes the IP address of to a DNS Blacklist.

As an experiment, I set up a sub domain and posted messages on Usenet. For each group I posted in, a unique email address was used to track which group the message originated. Within 2 days of posting a message, I'd received spam to these addresses!

```
Return-Path: <koralakeae@euskalnet.net>
Delivered-To: scott@yogi.scottclark.me.uk
Received: (qmail 21733 invoked by alias); 9 Feb 2004 14:29:46 -0000
Delivered-To: alt.abuse@spamtest.scottclark.me.uk
Received: (qmail 21730 invoked from network); 9 Feb 2004 14:29:45 -0000
Received: from lan-secretaria-de-la-reforma-agraria-d16-0206-0018.uninet.net.mx (HELO euskalnet.net)
(148.223.250.161)
  by yogi.scottclark.me.uk with SMTP; 9 Feb 2004 14:29:45 -0000
From: koralakeae@euskalnet.net
To: alt.abuse@spamtest.scottclark.me.uk
Subject: Test
Date: Mon, 9 Feb 2004 08:30:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary="-----=_NextPart_000_0011_DCCA620B.B2EF1E74"
X-Priority: 3
X-MSMail-Priority: Normal

This is a multi-part message in MIME format.

-----=_NextPart_000_0011_DCCA620B.B2EF1E74
Content-Type: text/plain;
  charset="Windows-1252"
Content-Transfer-Encoding: 7bit

The message cannot be represented in 7-bit ASCII encoding and has been sent as a binary
attachment.
```

The attached file was a virus, a copy of MyDoom-A.

That's how easy it is to get email addresses to send spam to, spam bots crawl websites and Usenet looking for email addresses in order to send spam to.

Advantages of Spam Blackholes

The advantages of Spam Blackholes are that once an email is received at one of these addresses the sending server can be added to a DNS Blacklist stopping it from sending any more messages.

Disadvantages of Spam Blackholes

I can't see any disadvantages to using blackholes in order to detect spam, they are important as they enable blacklists to be updated with computers that are sending unwanted mail.

Evaluation of Spam Blackholes

Spam blackholes are a good idea. They work in conjunction with DNS Blacklists to ensure that spam senders are discovered and blocked quickly. On my Mail Server, I have set up several email aliases that take email sent to them and automatically block the sender IP from sending any more. These aliases for addresses such as ceo@ admin@ and accounts@ have never been used for any of my domains, so I know that any mail sent to them is unsolicited and should be blocked.

Spam Filtering in Action

In this section of the report, different spam filtering tools will be investigated, to evaluate their effectiveness in detecting and filtering spam. How each individual test was carried out is described in the methodology.

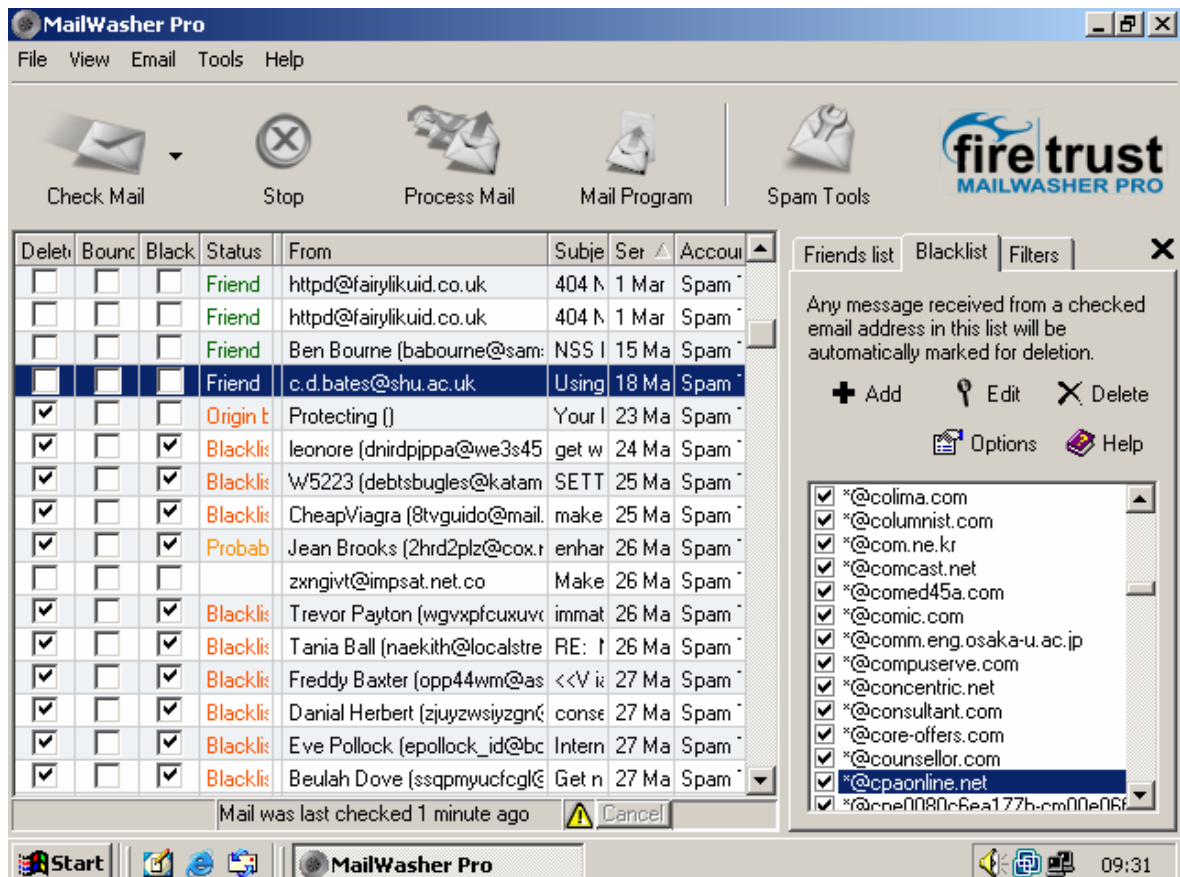
Software Review

At the start of this project, a search was done on the Internet to find off the shelf products that enable users to protect there inbox from the growing number of spam messages.

The following products were identified as being readily available; Mail Washer, McAfee Spam Killer, Norton Anti-Spam, Spam Assassin, Cloudmark and Spam Bayes. Before the results of the tests are discussed, the different products will be described as to how they achieve there goals.

Mail Washer

Mail Washer currently comes in 2 different versions, the free version and the Professional version, which was tested here. The free version is available from <http://www.mailwasher.net> whilst an evaluation copy of the professional version is available from <http://www.firetrust.com>. A licence for the professional version costs \$37. Mail Washer is available for Windows only.



The way that Mail Washer works is that it sits in front of your normal MUA; you check the mail there before you download it to your client. Once Mail Washer has downloaded your mail it analyses it to check for spam characteristics, it can also check DNS Blacklists and use your own personal friend/block lists. The key to ensuring that spam is discovered is to list the domains that you know you'll receive email from and the ones that it is highly unlikely to receive mail from. Once it has checked the messages it marks the messages that it suspects to be spam as such. Once you have selected your options, you click on the 'Process Mail' button, it deletes the messages you don't like and opens up your MUA to download the remainder of the messages.

Experiment Notes

As Mail Washer is a highly configurable piece of software it was included in the tests 4 times, each with a different set of configuration options. The different options are shown in the table below.

MW1	DNS Blacklists enabled (relays.ordb.org and sbl-xbl.spamhaus.org) Friends added to friend list Spam Addresses blacklisted.
MW2	As MW1 but with no DNS blacklisting
MW3	As MW1 but with no DNS blacklisting and instead of blacklisting the email address, the domain of the sender was blacklisted.
MW4	As MW3 but with DNS blacklisting.

After each test, the software was cleared of its black/white lists.

For the tests, all of the messages were copied into the mail directory on the Linux server and Mail Washer was allowed to download them as if it was downloading from your ISP.

If a message was missed as spam, it was marked as such and vice versa for wrongly marked clean mail.

McAfee Spam Killer

McAfee Spam Killer acts as a POP3 proxy. When you click on Receive in your MUA, the client is actually fetching the mail from McAfee. The system downloads mail in the background, allowing you to do other work.

The software is available from <http://uk.mcafee.com> and costs £20 plus a £17.99 annual subscription for software updates.

McAfee Spam Killer is available for Windows only.

The screenshot displays the McAfee SpamKiller application window. The interface is divided into several sections:

- Summary:** Overview of your SpamKiller status. Includes:
 - E-mail filtering is enabled:** [Click here to disable.](#)
 - Messages blocked today: 57:** [Click here to view.](#)
 - Friends List last updated: 27/06/2003:** [Click here to update.](#)
- Recent Spam:** Most recent e-mails that were identified as spam and blocked. A table lists the following entries:

From	Subject	Date	Rescue
"Jed Wilkins" <jedwilki...	soon to be?	13/04/2004 10:51	[Rescue]
"Shelly Maxwell" <eje...	acquire completelyfree sp0rts	13/04/2004 10:51	[Rescue]
"Download Cheap" <F...	Always be in the loop	13/04/2004 10:51	[Rescue]
Wallace Hammer <tbik...	acquire completelyfree sp0rts	13/04/2004 10:51	[Rescue]
"Marina Dupree" <hhq...	get kable at nocost	13/04/2004 10:51	[Rescue]
Mail Delivery Subsys...	Returned mail: see transcript for d...	13/04/2004 10:51	[Rescue]
"leonore" <dnirdpippa...	get with no hassle	13/04/2004 10:50	[Rescue]
- E-mail Overview:** Total e-mail received to date: 1800. Spam e-mail: 999. Spam (55%) [Progress bar].
- Recent Spam:** Spam received in the last 30 days. A pie chart shows the distribution of spam by category: Adult, Leisure, Financial, Products & Services, Security Threats, and Other.

The bottom status bar indicates: 0 accepted, 42 blocked. E-mail filtering is enabled.

When a spam message is discovered it keeps it locally, but doesn't allow you to download it to your MUA. Spam Killer uses a variety of techniques to recognise spam, although it's main method is pattern matching. The system is easy to use, allowing you to add addresses to your friend list by a simple click of the mouse. It doesn't use DNS blacklists and doesn't have the ability to use an address blacklist. If a message is wrongly classified as spam, you click the rescue button and it learns that the message is not spam and allows you to download to your MUA.

Experiment Notes

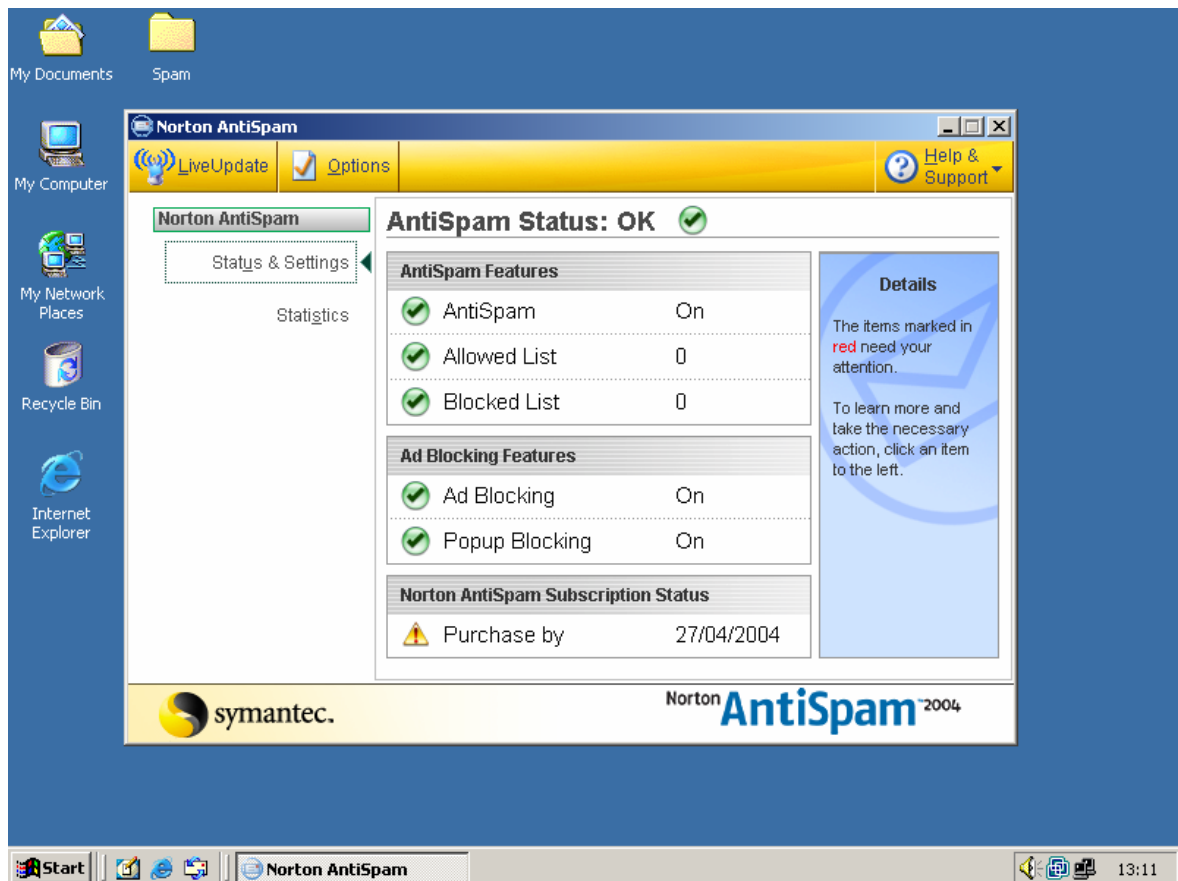
McAfee Spam Killer was tested using its default options. If a message was wrongly marked as spam, it was recovered and if a message was missed it will be marked as such. The friend list was used to specify which addresses that mail was legitimately received from.

For the tests, all of the messages were copied into the mail directory on the Linux server and McAfee Spam Killer was allowed to download them as if it was downloading from your ISP.

Norton Anti-Spam

Norton Anti-Spam acts in much the same way as a POP3 proxy, although it works at a lower level, examining all traffic sent via the POP3 and SMTP protocols – No changes to program configuration is needed to enable it to work. The program costs £29.99; a 30 day evaluation copy is available from <http://www.symantec.com>

Norton Anti-Spam is available for Windows only.



When a spam message is received, Norton Anti-Spam adds [Norton Anti-Spam] to the start of the message subject so they can be easily filtered out. The system can be trained on emails it gets wrong or that it missed. It does not include the facility to use DNS blacklists. Norton Anti-Spam allows you to create both white and black lists of addresses you either want to accept or reject mail from.

Experiment Notes

Norton Anti-Spam was due to be tested as part of the tests but during the download of the 1st batch of messages the system hung on the 92nd message downloaded. I'm not sure if it was due to something about the message or whether the software was broken. Because of this it was removed from the tests.

Spam Assassin

Spam Assassin is different from the other programs in this project; it works on the MTA, scanning messages as they arrive and not whilst you download them to your MUA. Spam Assassin is open source software; primarily running on Unix/Linux based operating systems, although being open source, and coded in Perl means that it has been incorporated into many different systems, some that run on Windows. Spam Assassin is available for download free of charge from <http://www.spamassassin.org>.

Being server based, it required a degree of technical knowledge to install and configure the software. The configuration files are text based.

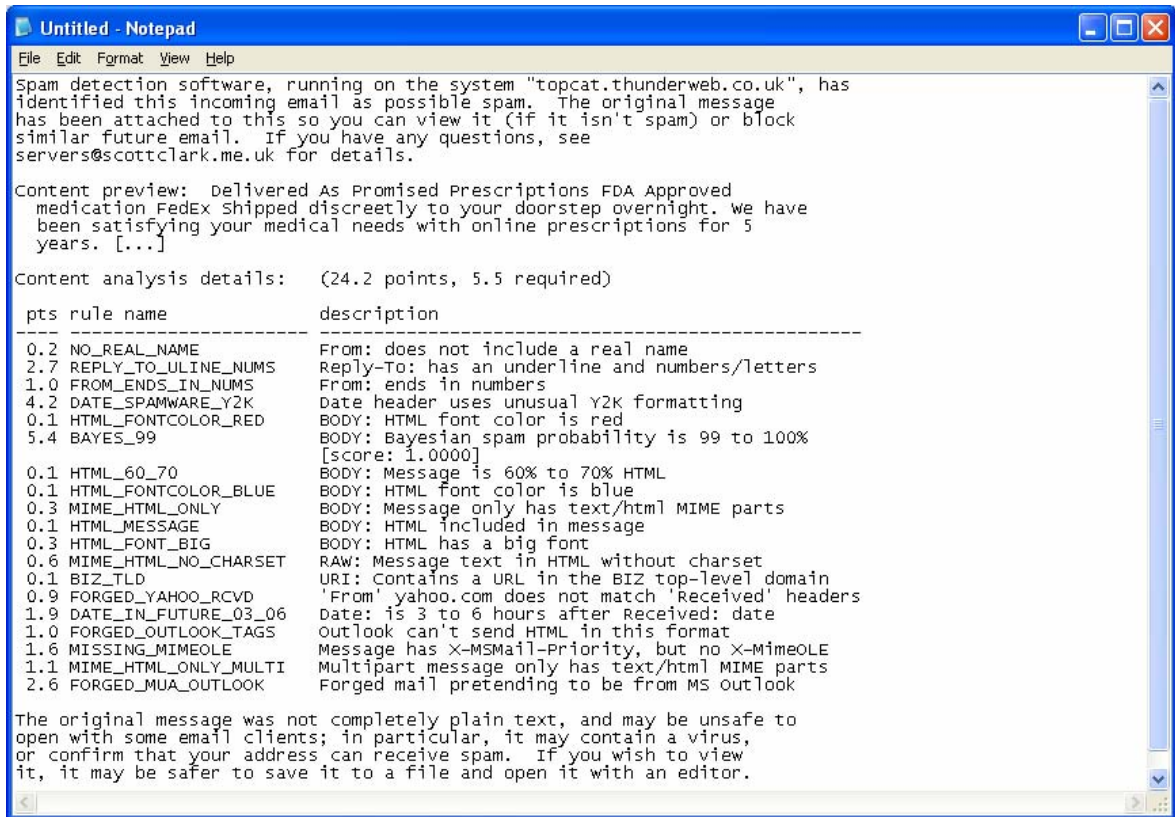
An example configuration file for Spam Assassin is shown below.

```
required_hits      5.5
rewrite_subject 0
report_safe       1
use_terse_report  0
auto_whitelist_path  /etc/mail/spamassassin/auto-whitelist
auto_whitelist_file_mode 0644
bayes_path        /etc/mail/spamassassin/bayes
bayes_file_mode   0644
use_bayes         1
bayes_auto_learn  1
auto_learn       1
skip_rbl_checks  0
use_razor2       0
use_dcc          0
use_pyzor        0
ok_languages     en
ok_locales       en
whitelist_from   mail@scottclark.me.uk
whitelist_from   o2Shop@o2.com
detailed_phrase_score 1
```

It is a massively complex piece of software but it is easy to configure, as you can see from the above, adding an address to the whitelist is as simple as typing it in.

The way that Spam Assassin works is that it gives points to certain characteristics of an email, for example it will give points if the message has forged headers or if any of the IP addresses in the header are in a DNS blacklist.

When a message is found to be spam, the message can either have an extra header put in, allowing your MUA to filter the mail, or rewrite the message as shown in the screenshot.



```
Untitled - Notepad
File Edit Format View Help
Spam detection software, running on the system "topcat.thunderweb.co.uk", has
identified this incoming email as possible spam. The original message
has been attached to this so you can view it (if it isn't spam) or block
similar future email. If you have any questions, see
servers@scottclark.me.uk for details.

Content preview:  Delivered As Promised Prescriptions FDA Approved
medication FedEx Shipped discreetly to your doorstep overnight. We have
been satisfying your medical needs with online prescriptions for 5
years. [...]

Content analysis details:  (24.2 points, 5.5 required)

pts rule name                description
-----
0.2 NO_REAL_NAME             From: does not include a real name
2.7 REPLY_TO_ULINE_NUMS     Reply-To: has an underline and numbers/letters
1.0 FROM_ENDS_IN_NUMS       From: ends in numbers
4.2 DATE_SPAMWARE_Y2K       Date header uses unusual Y2K formatting
0.1 HTML_FONTCOLOR_RED      BODY: HTML font color is red
5.4 BAYES_99                 BODY: Bayesian spam probability is 99 to 100%
                             [score: 1.0000]
0.1 HTML_60_70              BODY: Message is 60% to 70% HTML
0.1 HTML_FONTCOLOR_BLUE     BODY: HTML font color is blue
0.3 MIME_HTML_ONLY           BODY: Message only has text/html MIME parts
0.1 HTML_MESSAGE            BODY: HTML included in message
0.3 HTML_FONT_BIG           BODY: HTML has a big font
0.6 MIME_HTML_NO_CHARSET    RAW: Message text in HTML without charset
0.1 BIZ_TLD                  URI: Contains a URL in the BIZ top-level domain
0.9 FORGED_YAHOO_RCVD       'From' yahoo.com does not match 'Received' headers
1.9 DATE_IN_FUTURE_03_06   Date: is 3 to 6 hours after Received: date
1.0 FORGED_OUTLOOK_TAGS     Outlook can't send HTML in this format
1.6 MISSING_MIMEOLE         Message has X-MSMail-Priority, but no X-MimeOLE
1.1 MIME_HTML_ONLY_MULTI    Multipart message only has text/html MIME parts
2.6 FORGED_MUA_OUTLOOK      Forged mail pretending to be from MS outlook

The original message was not completely plain text, and may be unsafe to
open with some email clients; in particular, it may contain a virus,
or confirm that your address can receive spam. If you wish to view
it, it may be safer to save it to a file and open it with an editor.
```

Bayesian filtering is also used, it learns from all emails received, it can also learn from old email using a simple command line application.

A new version of Spam Assassin, version 3.70, should be released this year which adds support for checking SPF records.

Experiment Notes

Spam Assassin was tested twice, once using automatic Bayesian learning, the other using the '*sa-learn*' tool to teach the software what was and what wasn't spam.

Once the batch of mail had been tested, the following commands were issued to teach the system.

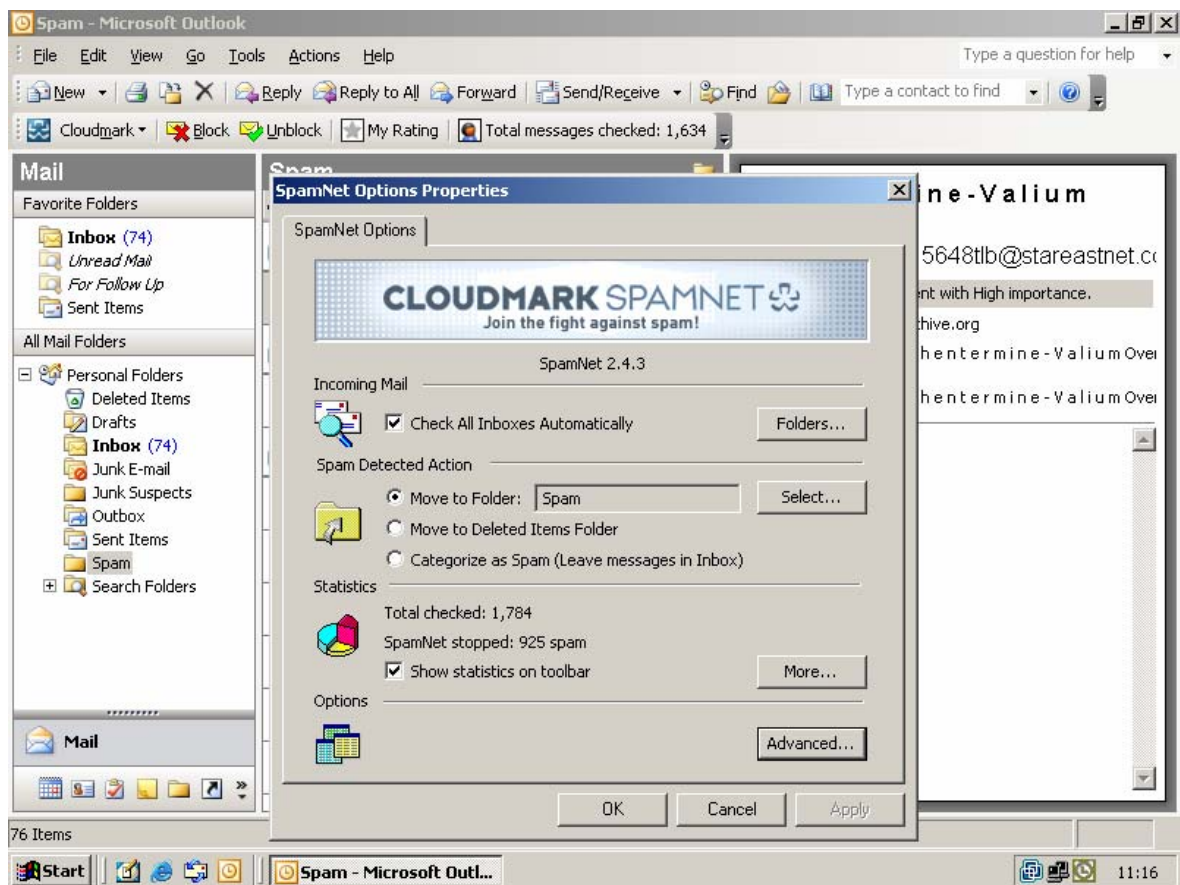
```
sa-learn -ham batch\clean  
sa-learn -spam batch\spam
```

As with the other experiments, each batch of mail was copied to the Mail directory on the Linux server, but as this software needed the messages to be received using SMTP the program 'Fetchmail' was used to download the messages and to send them back to the qmail install on that machine. To stop myself having to filter the messages with the spam header a script was written to do this as each mail was scanned. This script is shown in Appendix 5.

Cloudmark SpamNet

SpamNet is also different from the other programs used in this experiment; it uses a technique called ‘Collaborative Spam Fighting’. This means that when users receive spam they notify the Cloudmark’s server, uploading details of the message. This intelligence is then shared with other subscribers to the service and prevalent spam is automatically marked. The messages are still received by your MUA, but they are automatically filtered into your Spam folder.

Messages are nominated as spam or unblocked using buttons on the SpamNet toolbar in Outlook. Marking a message as spam sends its details to SpamNet.



SpamNet is available for download from <http://www.cloudmark.com>; a 30 day evaluation copy is available for download. A monthly subscription of \$3.99 is needed after these 30 days.

SpamNet can also learn from your existing mail, ensuring that it learns what is and what isn't spam.

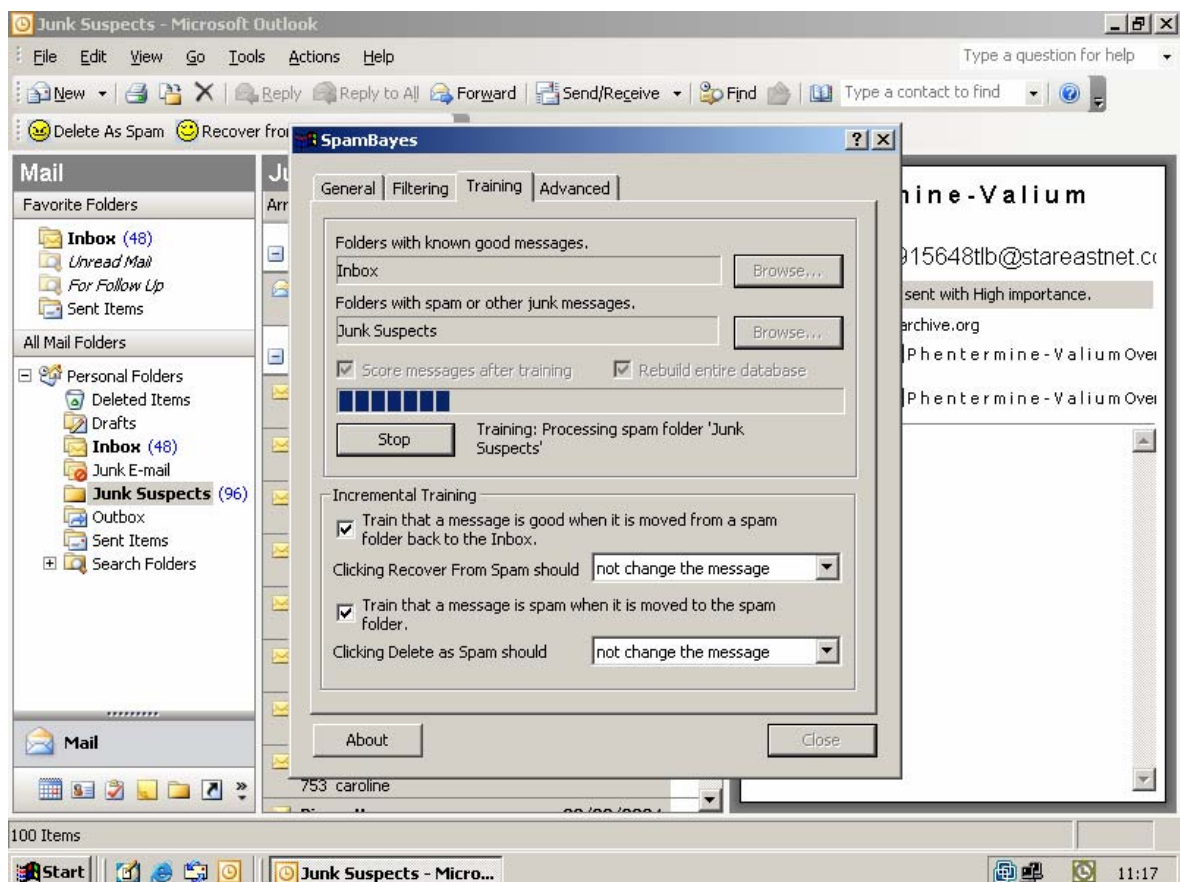
Experiment Notes

After each batch of emails was tested, those messages not marked as spam were selected and marked as spam, whilst email that was clean was unblocked. For the tests, all of the messages were copied into the mail directory on the Linux server and SpamNet was allowed to download them as if it was downloading from your ISP.

Spam Bayes

Spam Bayes, like SpamNet, works as an outlook plug-in. As the name suggests, the way it filters spam is by using the Bayesian techniques discussed earlier.

The first time the software is used, it moves ALL the messages into the 'Junk Suspects' folder, this is because it is supplied uneducated. You have to teach the software what is and what isn't spam. This is as simple as selecting the clean messages and clicking the 'Recover' button. On subsequent runs of the software, it was more intelligent, only moving the messages that it thought were spam.



Spam Bayes does not use DNS blacklists and does not have a whitelist feature. It only uses Bayesian filtering. Spam Bayes is free open source software, and is available from <http://spambayes.sourceforge.net>. Versions are available for both Windows and Unix / Linux desktops.

Experiment Notes

As stated above, on the first run all email was copied to the 'junk suspects' folder, all of the clean messages were selected and recovered. The 'Training' option was then used on both the inbox and spam folders to improve the accuracy. For the tests, all of the messages were copied into the mail directory on the Linux server and Spam Bayes was allowed to download them as if it was downloading from your ISP.

Test Results

As stated previously, each program was tested using ten batches of emails. The results are as follows. The results for each individual program are shown first, then a summary of all the results will be shown and comparisons made.

Mail Washer

The different conditions for each of the Mail Washer tests are shown in the table earlier in this report.

Test 1

	Possible	OBL	Blacklisted	Probably	Poss. Virus	Missed
Batch 0	3	84				13
Batch 1	2	82			3	18
Batch 2	2	70	2	1	5	15
Batch 3	7	97	3			13
Batch 4	19	53	2	4		34
Batch 5	13	37	2			25
Batch 6	32	32		1		27
Batch 7	9	83	1			40
Batch 8	12	93	2	1		32
Batch 9	10	41		1		18

OBL = Origin Black Listed (in a DNS Blacklist)

Test 2

	Probably	Possible	Blacklisted	Poss. Virus	Missed
Batch 0	2	17			81
Batch 1	2	14		3	86
Batch 2	2	20	2	5	66
Batch 3	2	33	3		82
Batch 4	4	31	2		75
Batch 5	2	23	3		49
Batch 6	2	39			51
Batch 7	6	28	1		98
Batch 8	3	36	2		99
Batch 9	2	16			52

Test 3

	Probably	Possibly	Poss. Virus	Blacklisted	Missed
Batch 0	2	17			81
Batch 1	1	9	3	20	72
Batch 2		14	5	19	57
Batch 3	2	27		34	57
Batch 4	4	22		35	51
Batch 5	2	13		23	39
Batch 6	1	34		22	35
Batch 7	3	15		40	75
Batch 8	3	18		51	68
Batch 9	2	10		19	39

Test 4

	Possible					Missed
	OBL	Possible	Blacklisted	Probably	Virus	
Batch 0	84	3				13
Batch 1	64	2	20			3
Batch 2	55	2	19			5
Batch 3	68	7	34			11
Batch 4	31	16	35	4		26
Batch 5	25	8	23	1		20
Batch 6	19	29	22			22
Batch 7	55	7	40			31
Batch 8	58	7	51	1		23
Batch 9	30	6	19	1		14

OBL = Origin Black Listed (in a DNS Blacklist)

McAfee Spam Killer

	Marked	Missed
Batch 0	69	31
Batch 1	103	2
Batch 2	94	1
Batch 3	105	15
Batch 4	93	19
Batch 5	62	15
Batch 6	71	21
Batch 7	94	39
Batch 8	97	43
Batch 9	43	27

Norton Anti-Spam

As stated previously, due to problems with this software it was dropped from the tests.

Spam Assassin

The different conditions for each of the Spam Assassin tests are outlined in the software review section of this report.

Test 1

	Marked	Missed
Batch 0	30	70
Batch 1	39	66
Batch 2	30	65
Batch 3	41	79
Batch 4	22	90
Batch 5	31	46
Batch 6	83	9
Batch 7	128	5
Batch 8	127	13
Batch 9	66	4

Test 2

	Marked	Missed
Batch 0	30	70
Batch 1	101	4
Batch 2	90	5
Batch 3	120	0
Batch 4	109	3
Batch 5	74	3
Batch 6	91	1
Batch 7	131	2
Batch 8	133	7
Batch 9	67	3

Cloudmark SpamNet

	Marked	Missed
Batch 0	75	25
Batch 1	85	20
Batch 2	65	30
Batch 3	89	31
Batch 4	84	28
Batch 5	58	19
Batch 6	59	33
Batch 7	100	33
Batch 8	107	33
Batch 9	60	10

Spam Bayes

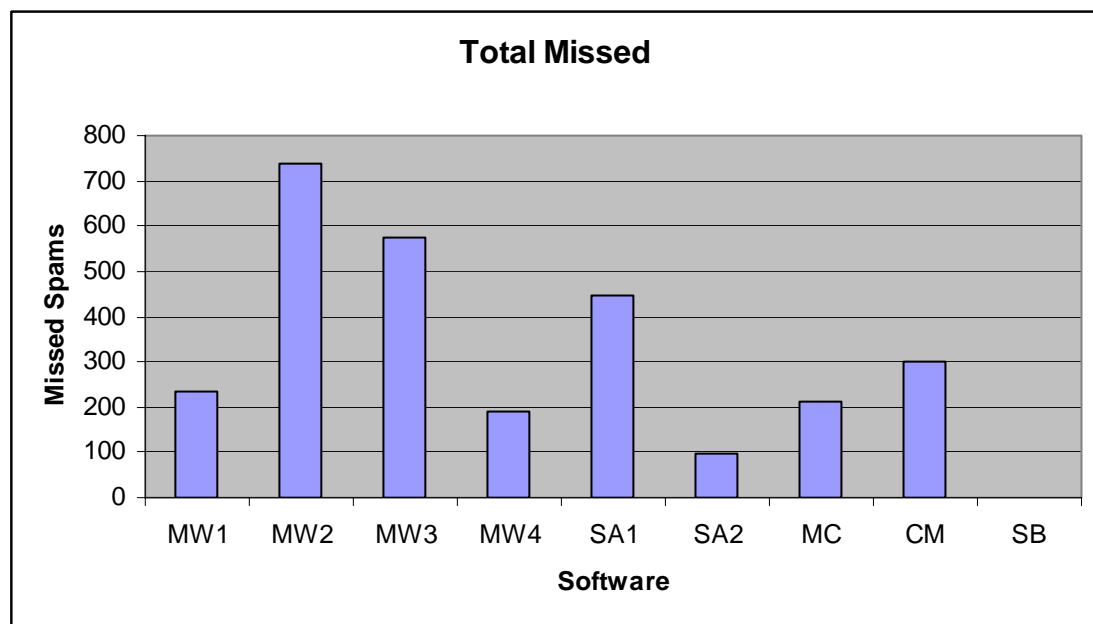
	Suspect	Marked	Missed	Wrong
Batch 0	150		0	50
Batch 1	4	101	0	4
Batch 2	3	92	0	7
Batch 3		120	0	1
Batch 4		112	0	1
Batch 5		77	0	1
Batch 6	1	91	0	1
Batch 7		133	0	1
Batch 8		140	0	0
Batch 9		70	0	12

Total Missed

	MW1	MW2	MW3	MW4	SA1	SA2	MC	CM	SB
Batch 0	13	81	81	13	70	70	31	25	0
Batch 1	18	86	72	16	66	4	2	20	0
Batch 2	15	66	57	14	65	5	1	30	0
Batch 3	13	82	57	11	79	0	15	31	0
Batch 4	34	75	51	26	90	3	19	28	0
Batch 5	25	49	39	20	46	3	15	58	0
Batch 6	27	51	35	22	9	1	21	33	0
Batch 7	40	98	75	31	5	2	39	33	0
Batch 8	32	99	68	23	13	7	43	33	0
Batch 9	18	52	39	14	4	3	27	10	0
Total Missed	235	739	574	190	447	98	213	301	0
% Missed	22.51	70.79	54.98	18.20	42.82	9.39	20.40	28.83	0.00

MW – Mail Washer, SA – Spam Assassin, MC – McAfee Spam Killer,

CM – Cloudmark SpamNet, SB – Spam Bayes



Price Comparison

Software	Price
Mail Washer (Pro)	\$37 ~ £22
McAfee Spam Killer	£19.99 + £17.99 Annual Subscription
Norton Anti-Spam	£29.99
Spam Assassin	Free
Cloudmark SpamNet	\$3.99 per month ~ £27 per year
Spam Bayes	Free

Software that costs more than another does not mean that it will perform better than a cheaper (or even free) package. Norton Anti-Spam was the most expensive package that was to be included in these tests, but it wouldn't even process the 1st batch of emails, all the other packages did!

Open Source software is free, yet from the results table you can see that they performed better than the software that had to be paid for. They may have a less 'polished' interface or be not as user friendly, but they do the job for which they were created better.

Discussion of Results

Just by looking at this tables and graph you would think that Spam Bayes was the best software at stopping spam messages getting into your inbox as it missed no spam messages.

On the first run it placed all the mail into the suspect folder, waiting for you to retrieve your good mail and mark all spam messages.

The package was trained using the first batch of emails. After the initial run, the majority of the spam was put into the spam folder; the few it missed were placed into the suspect folder.

As can be seen by the individual table, Spam Bayes gave some false positives, marking messages as spam when they were legitimate; this was the only piece of software on test that wrongly marked clean messages as spam.

Looking at all of the Mail Washer tests, you can see that without any DNS blacklisting much of the spam gets through. MW1 and MW2 use the same options apart from DNS blacklisting, with the addition of DNS blacklisting the amount of spam messages that are missed falls from 70.79% to 22.51%, that's over three times the amount of spam marked as such with the inclusion of DNS blacklisting.

The other Mail Washer tests, MW3 and MW4, continue this trend, more spam is blocked that during the first 2 sets due to the blacklisting of entire domains. This works because the envelope sender is normally forged on spam, so blocking entire domains from sending you mail is a good idea. The only problem with this is that if the domain is a popular one for both spammers to use and legitimate mail, such as hotmail.com, then you will block legitimate mail from this domain too. This did not come up in my tests because nobody from the popular web based email services, such as hotmail.com and yahoo.com, email me on a regular basis, so none were included in any of the batches of emails. This can possibly be seen as a flaw in the testing procedure. A lot of people do have accounts from these free email providers, so the majority of users will receive email from these services.

Looking at the 1st of the Spam Assassin tests you can see that after the batch 5 the number of spam messages missed is considerably reduced. Upon further investigation I found out that the Bayesian filter had been trained sufficiently and was correctly identifying messages that were spam.

The 2nd Spam Assassin test shows that the Bayesian filter works better if it is explicitly trained with known spam and clean messages, and not left to learn on it's own as in the first test.

Out of the box, McAfee Spam Killer seems to perform very well, it required little training. I think this was down to the regular updates to the spam database from McAfee; this is what you pay the yearly subscription for.

With Cloudmark SpamNet being a collaborative effort, I would have expected it to detect more spam messages than it actually did, surely I was not the only person to have received these spam messages, did nobody else report them? According to the statistics on Cloudmark's website they have 931,885 subscribers who use their system. Surely some of these messages would have been received by others?



Cloudmark Stats.

Overall, you can see a great difference in the amount of spam messages detected by the various software packages. With various options being chosen in some software to see what difference some of the technologies discussed in the earlier part of this report make, you can see that with the addition of DNS Blacklisting and Bayesian filtering increases the overall amount of spam detected. As stated previously in the report, there is not one key technology that can be used to reduce the amount of spam, a combination of different systems must be used together in order to reduce the amount of unsolicited commercial email that reaches your inbox.

Evaluation of Testing

The strategy used for testing the different packages was, I believe, to be a thorough method of testing the accuracy and effectiveness at filtering spam. The use of different batches of emails was a good idea, as it allows the packages to learn about the sort of messages that a user gets and how to recognise spam. This can be seen in the first Spam Assassin test, where after the 6th batch had passed through the package it had learnt about the different messages received and it became more effective in detecting which messages were and were not spam.

Two possible improvements could have been made to the testing process. The first is a wider selection of clean mail, including some received from people using free email providers such as hotmail.com or yahoo.com. This would have tested the packages to ensure that legitimate email from these systems was not wrongly marked as spam.

The second possible improvement to the testing process would have been to use each package in a 'real world' environment for a period of time. This would show how they worked on a vast variety of different messages and how effective they became over the course of 2 or three months.

Project Recommendations

When posting to Usenet, don't use your real email address.

- Put nospam or a similar word in there. E.g. scott@NOSPAMscottclark.me.uk, most people will know to remove the nospam and be able to contact you, robots won't; so they will try to send email to that address.

Buy a domain name with catch-all email facilities

- A domain name with catch-all email facilities can cost as little as £10 for 2 years. For each site that you register with, use a new email address so that you can trace your messages. For example, if you signed up with amazon.co.uk then you could use the email address amazon@scottclark.me.uk, thus any email coming to that address would be from Amazon. If spam were received to this address then you would know that Amazon had shared your details and you could filter that address.

Never click on the 'Unsubscribe' link in a spam message

- Clicking on an unsubscribe link in an email tells that spammer that this address is valid and that they should go on sending you more spam!

Turn off fetching remote images in your MUA

- If your MUA supports reading HTML messages, turn off the downloading of remote images. They can include images which track when you open messages, again telling the spammer that this email address is valid.

Use Anti-Spam software to filter your inbox

- If you already receive a lot of spam messages, ensure you use Anti-Spam software. The tests carried out as part of this report show that for a piece of software to be effective then it needs some sort of Bayesian filtering technology. The facility to query DNS blacklists helps an application to be more effective in knowing which messages are spam. Software that is recommended as part of this report is Spam Assassin or Mail Washer. Both are effective at deciding which messages are spam or not.
- Use white lists in your software to allow addresses which you trust to send you email.

Publish SPF Records for your domain

- If you have access to your DNS zone, add Sender Policy Framework records to say which hosts can send email from your domain. This will help prevent forgeries.

Don't put email addresses on a web page

- Don't include an email address on your web page. Spam bots crawl web sites looking for email addresses to add to their databases. Include a contact form on your site so that people can contact you.

If you must include an email address on your web page, make it more difficult for a spam bot to read.

- Using a simple piece of JavaScript code can hide your email address. By not including a standard mailto: link, a spam bot will not be able to get your address. A suitable piece of JavaScript could be:

```
<html>
<script language="javascript">
function SendMail(Login, Server)
{
    window.navigate("mailto:" + Login + "@" + Server);
}
</script>
<body>
    <a href="javascript:SendMail('mail', 'scottclark.me.uk')">Mail me</a>
</body>
</html>
```

- Using this script is as simple as including `Mail me` wherever you want your email address to appear.
- As you can see from the inside front cover of this report, I write my email address as mail at scottclark dot me dot uk, this sounds it out in phonetics, allowing a human to read the address but for a spam bot it will be more difficult to understand.

Ensure your MTA is not an open relay

- If you run your own MTA ensure that it is not an open relay. Use the tool found at <http://www.ordb.org> to test whether or not your server will send messages for third parties.

Ensure you have up-to-date virus / firewall protection

- The latest generation of viruses can have inbuilt SMTP servers so that once infected your machine becomes a machine that the spammer can use to send spam. Having up-to-date anti virus software on your machine can detect and eliminate these viruses, preventing your machine becoming a slave to the spammers. Having a firewall on your machine is also important, free software such as Zone Alarm (<http://www.zonealarm.com>) can tell you when a program is trying to access the internet, if the program is not authorised that it will be blocked.

Ensure your machine is free of Spy Ware

- Some pieces of software that you can download include 'additional' programs that are supposed to add new features to your system. Quite often they programs are actually Trojan horses, they are meant to do one thing while they actually do another. They can act as SMTP servers turning your machine into a spam machine for the spammers. Most firewall and anti virus packages should be able to detect a Trojan. Anti Spy software is available on the Internet to remove these packages.

Evaluation of Recommendations

Most of these recommendations are common sense.

In the current climate, it is not a good idea not to have your computer protected from viruses and if you are connected to the internet then a firewall should be compulsory. Companies have seen the problem that reading an email can do, Microsoft in the latest version of Outlook for example have turned off by default the remote download of images which could track when a user opens an email.

The recommendation of buying a domain name and using a unique address for each site that you use is one that a growing number of people are doing; it allows the MUA to sort mail into folders and allows the user to find out where an email originated from.

If a user adopts some of these recommendations, they should notice a drop in the amount of spam that is received.

Practical Implications and Further Development

At the start of this project I had a major spam problem. I would receive in excess of 450 messages a day; the majority of them would be spam. At this time I was using just one email address, `scott@scott-clark.co.uk`, this domain had been registered since May 2000, approx. 4 years. I took the drastic step of blocking this address completely, having it bounced at the RCPT TO: stage of the SMTP conversation. This also completely eliminated the spam that I received. Machines still try to send email to this address, between 1st February and 31st March 2004 approximately 5300 messages were sent to `scott@scott-clark.co.uk`, none of these reached my inbox.

I updated my contact details on websites that I had registered on, so that each site had its own unique address; this allows me to track where messages originate from.

This helped recently when I started receiving the occasional spam message. Running Spam Assassin on my MTA marked these messages as spam, but I still wanted to find out where they were coming from and to block them.

I tracked down which website had shared my details, closed my account with them, and blocked the address.

As part of the recommendations, it was suggested that using Anti Spam software with a DNS blacklisting facility would be a good idea. It is, but it all depends how it is implemented. If the check is done at the SMTP conversation level, where an IP address can be blocked from sending completely then the user might not receive an important message. This could be due to a MTA being wrongly configured or if the RBL blocked an entire network of addresses, not just a single IP. If this were the case, and an important message was blocked the recipient might not find out about the message until the sender tries to contact them via another means.

The way in which a blacklist should be used in my opinion is with a software package that uses DNS blacklists for scoring purposes, whereas a message from an IP on a blacklist would receive a higher score than a message from an IP address not on the list. This ensures that the recipient receives all mail addressed to them, they can then decide what to do with the message.

If more time was available to develop this project further, I would research more possible solutions to filtering spam. Technologies such as a 'Challenge / Response' system and collaborative filtering techniques such as Razor would be at the top of my list of systems to research.

I would also like to find out why Norton Anti-Spam didn't like one of the emails that I tried to pass through it. It would be interesting to note the conditions of this email, the headers and content that caused the program to hang. I'm sure the vendor, Symantec, would like to see this email so that they can update the software so that it doesn't hang the system.

A week after I completed the Spam experiments, a new version of Mail Washer was released. If the time was available, I would like to compare the new version with the old one to check on the updates and to see if an already effective piece of software has been made even more so.

The June 2004 edition of 'Computer Shopper' magazine included a group test of different Anti-Spam tools, unfortunately this magazine was received in the week prior to the submission deadline of this report. If more time was available I would have liked to investigate the additional tools mentioned in the article. Tools such as SpamPal, Aladdin SpamCatcher, Lyris MailShield Desktop and SpamStopUp would have made the scope of the tests broader, and seen which of these packages are the most effective and whether or not the conclusions of the article agree with mine or not.

During the tests, the Bayesian filters were trained after each batch of emails had been scanned. If the tests were to be completed again, I would like to use the software in real world conditions for at least 2 months. This would test their effectiveness over a period of time, not in the test conditions that they were subjected to for this report.

Another idea that could have been investigated further is the possible inclusion of a volunteer based test, where volunteers use a particular package for a period of time, for example 2 months, on their incoming email. This would have given an insight into how the packages performed in a real world environment and not in laboratory tests.

I would have also liked to have found out whether or not many domains had adopted 'Microsoft Caller-ID for Email'. During my investigation, I only found reference to three domains having published records, these being Microsoft.com, Hotmail.com and Amazon.com, 2 of these are Microsoft sites. I would have thought that Microsoft would be pushing the technology and more organisations would be publishing records.

Client Contact and Evaluation

During the course of this project, regular contact had been kept with my supervisor.

Meetings were arranged and attended and regular email updates were sent, informing my supervisor of the overall status of the project.

As part of these meetings, a member of CIS staff was made available, and questions of the Sheffield Hallam University mail system were posed.

Regular contact has also been made with the company I went on placement with, TechnoPhobia.

Having contact with organisations that use anti-spam systems has given me an insight into what can be done to help detect and filter spam messages. It has also helped me to understand the different contexts in which both organisations are in.

One is in the educational sector whilst the other is in the Web Design industry. With this in mind, they have differing needs in terms of what they require from a mail system. A university's mail system might not be mission critical, but for a medium size business in the technology sector the mail system will be an important aspect of the company, with details about projects for tender coming in they can not afford to have a mail system that does not accept all mail properly.

Ideally, I would have liked this project to have a 'real world' client, this would have enabled me to improve my communication skills and would have allowed be to liaise with a third party in order to gather their specifications and requirements. This would have allowed me to study and adhere to the British Computer Society's code of conduct.

As part of the software trials, I would have liked to have had some volunteers that would have used a particular software package for a period of time to see how effective they were in a real world environment. I could then have kept in regular contact with the volunteers providing insight into the packages.

Progress against Project Plan

At the start of the project, a plan was created to allow me to plan my time. The aim was to make effective use of my time in order to achieve the aims listed in the objectives.

As part of my university course, it was taught that the effective planning of time would enable a project to be completed on time, without a last minute rush to get things completed.

As part of the project specification, the following task plan was created, along with the envisaged number of days that would be needed.

1	Produce a project plan	1 day
2	Investigate different ways that spammers get your email address.	2 days
3	Investigate new ways of preventing spam	10 days
4	Research the offering of products currently available.	2 days
5	Devise a method of testing the different products	2 days
6	Compare and contrast the different products	6 days
7	Identify suitable software to use in prototype system.	Same as 4
8	Build prototype system (VMWare Images)	1 ½ days
9	Produce a set of recommendations to help prevent spam and virus threats.	2 days
10	Evaluate the testing strategy used.	1 day
11	Critically evaluate the project overall	2 days
12	Produce a report on the execution of the project.	20 days

For the most part, these time scales were adhered to, the 20 days for the project report was a wild under estimate. Sections of the report were written throughout the lifetime of the project. It is estimated that approximately 35 days was used to create the report. In the above plan this can be accounted for as an ongoing task, with time from each task being used to add to the report on a specific section.

As well as the task plan, a plan for what work needed to be completed in which week was also created. The plan for the weeks between December 2003 and April 2004 was as such:

Week Beginning	Key Project Ideas
01/12/2003	Discuss Spec with Tutor / Refine
08/12/2003	Research Anti-Spam Solutions
15/12/2003	Research Anti-Spam Solutions / Questionnaire
22/12/2003	* Christmas * - Week Off
29/12/2003	Research Spam Harvesting Methods
05/01/2004	Other Research (Software)
12/01/2004	Context Research
19/01/2004	Start Project Write up - Solutions
26/01/2004	Solution Write up
02/02/2004	Solution Write up
09/02/2004	Solution Write up
16/02/2004	Complete Anti-Spam Write up
23/02/2004	Gather Email for Tests
01/03/2004	Spam Filtering Tests
08/03/2004	Spam Filtering Tests
15/03/2004	Analyse Results
22/03/2004	Write Up
29/03/2004	* Weekend Away * - Write Up
05/04/2004	* Easter * - Write Up
12/04/2004	Ideas for Improvement
19/04/2004	Critical Evaluation
26/04/2004	Proof Reading / Finalising / Binding

Right from the very start this plan was not adhered to strictly. Feedback on my project specification was due at the end of November 2003, I did not receive any proper feedback until the start of January 2004, this lost me a months worth of time that could have been spent starting the writing up. This month was not completely wasted though; I spent the time doing background reading and researching different software methods that could be used.

After this problem, the idea was to fit additional research and the start of the project write-up around my other university assignments. Unfortunately the majority of assignments in the final year were based in the second semester. I found the work load between February and March to be very demanding on myself. I was working long days on other assignments and had to put my project on the 'back burner', thinking about it and researching different methods when I had spare time.

By the end of March 2004, the workload had cleared and time was made available to work on the dissertation. By the end of Easter, the majority of the report had been completed; the experiments had been completed and the different methods of spam prevention had been written up.

As can be seen from the plan, the critical evaluation was due to be written in the week prior to submission. Unfortunately I caught a viral infection and was unable to work to my full potential. This lack of work for 4 days meant that the plan had fallen behind again. This meant that the evaluation had to be written in the days before submission, the time that was planned for finalising the report and proof reading it.

Due to the late project feedback, the rush of work in the second semester and being ill, the end of the project seemed too rushed to me. If I were to go back and do the project again I would have tried to do more research and testing before the Christmas period, when my work load was less. This would enable me to spend the last month writing up, ensuring that all the points that I wanted to cover had been covered. I would have also liked to the time to work on some of the items that were covered in the 'Further Development' section of this report.

Critical Discussion of Own Work

Through this report there has been evaluation at different stages, from evaluating the different anti-spam solutions to evaluating the testing procedure used. This section of the report discusses the way I worked throughout the lifetime of the project, both the good and the bad things that happened during the project and what I learnt, both in terms of spam filtering and working on large projects. It will also evaluate some of the sources used, to see if they are adequate.

At the start of the final year, I didn't realise the amount of work that would be given in addition to the final year project. The 1st semester I tended to take things easy, working on assignments when they were almost due. No advance planning was undertaken to ensure that work could be done at a steady pace throughout the year. In hindsight, I would have liked to have completed more initial research at the start of this academic year. Having worked in industry for the year prior to returning to university it took me a considerable amount of time to get back in 'with the flow', to settle back in and complete work on schedule.

During the course of this project I have learnt how to manage my time more effectively. This improved over the course of the year; it ensured that even though I was ill in the week prior to submission, I still had enough time to complete the project on time. I have also learnt about different ways of researching topics. My primary source of information was various sites on the Internet, although some paper based sources were used effectively.

In Woods report entitled 'A Spammer in the Works', he talks about the problems of spam and the cost to business about not having some mechanism in order to reduce the amount received. While this information is valuable as it puts the spam problem in a business context, it needs to be remembered that this was written by a company that provides anti-spam services; they have a vested interest in painting a bleak picture of the spam problem and how much it will cost you business if you don't act now, as they are trying to see you a service. Even with this in mind, the information presented in the paper is a valuable resource, I know from first hand experience of the scale of the spam problem, and the cost to business.

I believe that the other sources referenced in this report show a broad and balanced view of the spam problem, Woods' report does as well and it is from a reliable company so the information presented can be trusted. Microsoft's documentation on 'Caller-ID for Email' suggests that their solution is the end of the spam problem, and that everybody should rush out and adopt it. This is a bit one sided as it is only part of the overall solution. Differing opinions exist as to whether or not it will be effective. It is still to be proved whether the system will reduce the amount of spam messages received, with only a handful of domains publishing records then only when a greater adoption has been achieved can this be tested.

At the start of the project, I had planned on putting out a questionnaire in order to gather people's reactions to spam, and what they do to combat the growing problem. This questionnaire is included in Appendix 6. I received several responses from friends, the results gathered did give an insight into other peoples' experiences of spam, but the questions posed did not really lead to any results that could be evaluated. For this reason the results from the questionnaire are not included in the report. What should have been done was to create a set of questions that were focused on a specific area, with results that could have been analysed properly. It could have gave a good insight into others experience of spam.

Overall the process of setting up and completing a project of this scale has been beneficial. It has enabled me to improve various skills, from report writing and research methods, to time management and improved communication skills.

The skills and methods learnt during the past 6 months will put my in good stead for working life after graduation.

At first the idea of completing a project of this size was daunting; I'd never completed a piece of work this large before. But once I'd sat down and planned what I was going to do, the worries went away. Once my time management skills had improved, the deadlines were kept too and the project was completed on time.

References

Association for Computing Machinery, ACM Code of Conduct, Last accessed on 30th March 2004 at URL: <http://www.acm.org/constitution/code.html>

BBC News, Sobig is biggest virus of all, [Online], Last accessed on 31st December 2003 at URL: <http://news.bbc.co.uk/1/hi/technology/3169573.stm>

British Computer Society, BCS Codes of Conduct and Practice, [Online], Last accessed on 30th March 2004 at URL: <http://www1.bcs.org.uk/link.asp?sectionID=673>

Brad Templeton, Reaction to the DEC Spam of 1978, [Online], Last accessed on 22nd March 2004 at URL: <http://www.templetons.com/brad/spamreact.html>

John Postel, (1975), On the Junk Mail Problem – RFC 706, [Online], Last accessed on 30th March 2004 at URL: <ftp://ftp.rfc-editor.org/in-notes/rfc706.txt>

Message Labs, (2003), December 2003 Monthly View, [Online], Last accessed on 19th February 2004 at URL: <http://www.messagelabs.com/binaries/Dec03.pdf>

Microsoft, (2004), Caller-ID for Email – The Next Step to Deterring Spam, [Online], Last accessed on 31st March 2004 at URL: http://www.microsoft.com/mscorp/twc/privacy/spam_callerid.mspcx

SpamHaus, Frequently Asked Questions, [Online], Last accessed on 1st April 2004 at URL: <http://www.spamhaus.org/sbl/sbl-faqs.lasso>

SPF Registry, Early Adopters, [Online], Last accessed on 1st April 2004 at URL: <http://spftools.infinitepenguins.net/register.php>

Tschabitscher, Heinz, What You Need to Know About Bayesian Spam Filtering, [Online], Last accessed on 2nd April 2004 at URL: <http://email.about.com/cs/bayesianfilters/>

Wood, Paul (2003), A Spammer in the Works, [Online], Last accessed on 16th April 2004
at URL: <http://www.messagelabs.com/binaries/A%20spammer%20in%20the%20works.pdf>

Bibliography

Anti-Spam Resources and Tools, Doug Bagley, May 2003.

<http://www.bagley.org/~doug/spam/resources.shtml>

Computer Shopper, June 2004 Edition, Dennis Publications, Anti-Spam Software Lab,
Pages 225 – 229

Fighting Spam for Dummies, JR Levine, John Wiley & Sons Inc., February 2004

ISBN 0764559656

Guide to Spam Reduction, Don McKenzie, e-dotcom.com, March 2004.

http://www.e-dotcom.com/spam_exp.php

Personal Computer World, December 2003 Edition, VNU Business Publications, Spam
Solutions, Pages 167 – 175

Protecting your website's email addresses from being used by spammers, Aron Roberts,
University of California, 2003.

<http://istpub.berkeley.edu:4201/bcc/Winter2003/feat.spamharvest.html>

Removing the Spam: E-mail Processing and Filtering, Geoff Mulligan, Addison Wesley,
May 1999.

ISBN: 0201379570

Second-Generation Anti-Spam Solutions, Kaitlin Duck Sherwood, September 2002.

<http://www.overcomeemailoverload.com/advice/AntiSpamTools.html>

Sender Policy Frame Documentation, Sender Policy Framework Group, April 2004.

<http://spf.pobox.com/>

Spam Links, January 2004.

<http://spamlinks.port5.com>

Spam Percentages and Spam Categories, Brightmail Ltd. April 2004.

<http://www.brightmail.com/spamstats.html>

Stopping Spam: Stamping Out Unwanted Email and News Postings, Alan Schwartz and Simon Garfinkel. O'Reilly UK. October 1998.

ISBN: 156592388X

Stopping Spambots, Neil Gunton, December 2003.

http://www.neilgunton.com/spambot_trap/

'The Spam problem and how it can be countered', The National Office for Information Economy (Australian Governmental Body) report. August 2002,

http://www2.dcita.gov.au/__data/assets/file/13050/SPAMreport.pdf

The European Coalition Against Unsolicited Commercial Email, EuroCAUCE, January 2004.

<http://www.euro.cauce.org/en/>

The Harvester Project, SpywareInfo, January 2004.

http://www.spywareinfo.com/harvest_project/

Yahoo! Anti-Spam Resource Center, Yahoo! Inc. April 2004.

<http://antispam.yahoo.com/>

Appendix 1 – Project Specification

Project Definition

Student: Scott Clark

Date: 13/10/2003

Amended: 01/02/2004

Supervisor: Samir Alkhayatt

Level of project: BSc (Hons) Networks and Communications

Title of Project: Spam Detection and Filtering Techniques

ELABORATION

How many times have you opened your email and been inundated with emails advertising penis enlargement, Viagra and Weight loss pills? The problem of unsolicited commercial email (Spam) is growing. In August 2003, the company MessageLabs, who are a provider of email security services to businesses, intercepted more than 91.2 million spam emails¹. They say that this is a global spam ratio of 1 in every 2.8 emails.

August 2003 also saw a massive amount of email borne viruses. The outbreak of Sobig.F was reported to be one of the fastest growing viruses ever². MessageLabs reported that in the first 24 hours of the virus outbreak, more than one million copies of the virus were detected.

The need for comprehensive scanning of emails is extremely important. This project aims to compare and contrast the different ways of stopping the threat of viruses and the inconvenience of spam emails. This will involve looking at existing systems, both on the SMTP server and on the desktop. After the products have been evaluated, a report and an example system will be produced.

¹ Source - <http://www.messagelabs.com/binaries/Aug03.pdf>

² Source - <http://news.bbc.co.uk/1/hi/technology/3169573.stm>

PROJECT OBJECTIVES AND DELIVERABLE

The student is required to:

Explore and understand the different methods and tools available to undertake this project.

Explore how email addresses are harvested by spam senders.

Research different methods of filtering / scanning email both on the server and desktop.

Using a methodical approach, test the different systems and compare features etc.

Research different methods of trying to prevent unsolicited emails

Produce a set of recommended methods to prevent such threats.

Critically evaluate the recommendations produced and evaluate their usefulness.

Critically evaluate the usefulness of the methods and tools used to create the recommendations and illustrate any drawbacks found

Suggest future enhancements and development of the recommendations produced

Produce a report of the project.

The deliverable for this project will be a report outlining a set of recommendations to help prevent spam and virus attacks.

TASK PLAN

- 1 – Produce a project plan
- 2 – Investigate different ways that spammers get your email address.
- 3 – Investigate new ways of preventing spam
- 4 – Research the offering of products currently available.
- 5 – Devise a method of testing the different products
- 6 – Compare and contrast the different products
- 7 – Identify suitable software to use in prototype system.
- 8 – Build prototype system
- 9 – Produce a set of recommendations to help prevent spam and virus threats.
- 10 – Evaluate the testing strategy used.
- 11 – Critically evaluate the project overall
(Were all the deadlines met, did everything go according to plan?)
- 12 – Produce a report on the execution of the project.

Appendix 2 – The 1st Spam Message Sent

This is a copy of the first spam message ever sent over Arpanet. Most of the addresses have been left out of this copy as there were 5/6 pages full! Brad Templeton includes the full message, including all the addresses, on his homepage. He writes that the reaction was ‘not unlike the reaction to spam today.’

Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT

Date: 1 May 1978 1233-EDT

From: THUERK at DEC-MARLBORO

Subject: ADRIAN@SRI-KL

To: DDAY at SRI-KL, DAY at SRI-KL, DEBOER at UCLA-CCN,

To: WASHDC at SRI-KL, LOGICON at USC-ISI, SDAC at USC-ISI,

To: DELDO at USC-ISI, DELEOT at USC-ISI, DELFINO at USC-ISI,

<Snip addresses>

DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM AND THE DECSYSTEM-10 <PDP-10> COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM. THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040 AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE DECSYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER DECSYSTEM-20 MODELS.

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 - 2 PM

HYATT HOUSE (NEAR THE L.A. AIRPORT)

LOS ANGELES, CA

THURSDAY, MAY 11, 1978 - 2 PM

DUNFEY'S ROYAL COACH

SAN MATEO, CA

(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY.

Appendix 3 – SPF Syntax

There are other features of SPF that allow the system administrator to specify which hosts can send email. They can specify that the MX servers can send email as well as receive, that any PTR record that ends with the domain name is valid and you can include your ISP's domain name if they may send email with your domain name.

In the following SPF record for example.com:

```
"v=spf1 a mx ptr ip4:82.68.38.136/29 a:www.example2.com mx:isp.com -all"
```

The v=spf1 specifies the version of SPF used.

a specifies that the A record for example.com can send mail.

mx specifies that the MX servers for example.com can send mail.

ptr specifies that any host whose reverse IP is in the example.com domain can send mail.

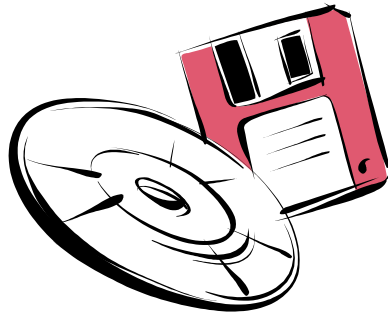
ip4:82.68.38.136/29 specifies that any host in the range 82.68.38.136-82.68.38.143 can send mail.

a:www.example2.com says that the host www.example2.com can send mail.

mx:isp.com specifies that the MX servers for isp.com can send mail.

-all specifies that this record encompasses all the hosts that can send mail for example.com

Appendix 4 – VMWare Images / Spam Messages



Appendix 5 – Spam Filtering Script

This script is run via a .qmail file, it takes a username as a parameter, checks whether the X-Spam-Status: Yes header is set, if so it moves it to the spam folder, if not it puts it into the new folder.

Then a simple `l message.* | wc -l` in each of the folders tells how many emails were classed as spam or clean.

```
#!/bin/sh

datetime=`date +%Y%m%d%H%M%S`

case $# in
0)    echo "Usage: $0 username" >&2
      exit 1
      ;;
*)    username=$1

      cat > /tmp/message.$datetime.$$tmp || exit 1

      if [ "`cat /tmp/message.$datetime.$$tmp | grep "X-Spam-Status: Yes"``" ] ; then

      echo This Message Is Classed As Spam By SpamAssassin
      echo Moving Message To $username/Maildir/.Spam/new
      mv /tmp/message.$datetime.$$tmp /home/$username/Maildir/spam

      else

      echo This Message is Assumed Clean
      echo Moving Message To $username/Maildir/new
      mv /tmp/message.$datetime.$$tmp /home/$username/Maildir/new

      fi

      exit 0
      ;;
Esac
```

Appendix 6 – Spam Questionnaire

scott clark

questionnaire

questionnaire

Fields marked with a * are compulsory, whilst those marked with # should only be completed if they are applicable to you.

Your name: *

Your email:

Location:

Company Sector: #

No. of Employees: #

No. of emails you receive per week: *

No. of these junk / spam: *

Time spent sorting junk from normal mail: *

No. of mails your mail server handles per week: #

No. of these junk / spam: #

Mail Server Used: #

What do you do to combat spam: *

Please give some examples of recent spam: *

Have you ever tried to reply to a junk email to try to get yourself removed?
(Please give details):

Any other comments / points that you'd like to make:

Send it!

last updated: January 14th 2004

© [Scott Clark](#) 2004.