The problems associated with operating an effective anti-spam "blocklist" system in an increasingly hostile environment.

Robert Gallagher
MSc in Security and Forensics
School of Computing
Dublin City University

August 10, 2004

1 Background

1.1 History of Spam

The earliest known email that could be classified as spam was sent in April 1994 by the law firm Canter & Siegel, advertising their services to those wishing to take part in the US government lottery of green card work permits [Templeton, 2003]. Around six thousand copies of this message were posted to Usenet discussion forums, breaching the unwritten rules of 'netiquette' that had governed behavior on the newsgroups up until that time.

The terms 'spamming' and 'spam' had been coined to describe widespread and unwanted postings to Usenet newsgroups, which would usually be unrelated to the topic being discussed. The term spam is a reference to a Monty Python sketch in which spam is the main ingredient of every dish in a cafe.

1.2 The Real Cost

Ten years later, in April 2004, spam accounted for 67.6% of 840 million messages assessed by the security firm MessageLabs[1]. The combination of an enormous potential audience and the ease of reaching that audience has made email into a very attractive medium for marketing, scams, politics and religion. Legitimate businesses and users have paid the price however. In a study conducted by the Radicati Group it is estimated that deploying extra infrastructure to deal with spam cost companies around the world 16.7 billion euro in 2003 [Radicati, 2003].

1.3 Simple to Sophisticated

Attempts have been made to reduce spam both at the client and server levels, from simple keyword filters to bayesian filters and blocklists - examples of which are described in section 2.2. These techniques have grown more sophisticated as the volume of spam has increased, but unfortunately the tactics employed by spammers have become just as ingenious. This has lead to what some have called the spam 'arms race'.

Spammers have begun to enlist the services of malware authors in order to create viruses and worms that aid in the distribution of spam, usually with the purpose of concealing its origin. In section 3 the growing links between spammers and malware authors are illustrated, which is one of the main goals of this practicum.

2 Blocklists

2.1 Background

Blocklists are simply databases that contain the IP addresses of known spam operations or computer systems that can be exploited to send spam[2]. Most modern SMTP servers can be configured to

[1] http://www.theregister.co.uk/2004/05/25/spam_deluge/

[2] These systems may be machines compromised with a virus or SMTP servers that do not place restrictions on the relay of email - 'open relays'

query a blocklist on receipt of an email message, extensible filters such as SpamAssasin can also make use of blocklists.

If the source IP of the message (retrieved from the email headers) exists in the database, the server can discard the message altogether or flag it as possible spam - in this case it is up to the client side email application to deal with the message.

ISPs, educational institutions, businesses and government agencies have all made extensive use of blocklists. Many blocklist systems have come into being since the earliest, MAPS RBL, began operation in 1996. Some of these systems are free, others are partly subscription based providing extra services, more comprehensive filtering and support. Because of their popularity, many blocklists have become the target of attacks, these are described in section 3.1.

The actual inclusion of an IP in a blocklist usually requires some human intervention in the form of a review process. Most blocklists encourage members of the public to submit spam sources through a well defined procedure, this minimises the amount of false positives that appear in the blocklist.

2.2 Current Blocklists

2.2.1 Spamhaus

The spamhaus project is one the better known blocklist systems. Spamhaus provides several services:

- SBL (Spamhaus Block List) - A real time database of IP addresses associated with known spam sources. Email servers can easily be configured to query the SBL on receipt of a message, and discard it if it comes from a verified spam source.

- XBL (Exploits Block List) - XBL is similar to SBL, except it stores the IP addresses of 3rd party exploits such as open proxies and malware[3] designed to aid in the distribution of spam (worms and viruses).

- ROKSO (Register Of Known Spam Operators) - A database that stores information and evidence on known spam operations.

Spamhaus' widespread use by ISPs and other organisations led to it falling under successive DDoS attacks throughout 2003 due to the Mimail, Fizzer and SoBig worms. This, and other attacks against blocklists are described in section 3.1.

2.2.2 Spamcop

Spamcop began as a spam notification and reporting system. Emails reported to SpamCop are analysed to determine who originally sent them and any email addresses or web site URLs in the body of the mail. The SpamCop system then contacts the relevant system administrators to inform them about the problem.

The reporting service quickly gained popularity and SpamCop began to offer commercial email accounts, site-wide corporate filtering and a blocklist service which solicits donations. However the blocklist that SpamCop operates has not been very successful. The listing process that the SpamCop blocklist employs appears to result in large numbers of legitimate IPs being incorrectly listed, in fact SpamCop actually warns against use of its blocklist in a production environment.

2.2.3 MAPS RBL

The MAPS RBL (Realtime Blackhole List) is a commercial blocklist that began operation in 1996, making it one of the earliest anti-spam systems. Comprehensive guidelines have been formulated in regard to how sources of spam are to be reported, and what constitutes a spam source. MAPS offers several other IP address listing services that do not necessarily list known sources of spam, but systems that could be used to send spam due to poor configuration - usually open relays or open proxies.

2.3 Alternatives to Blocklists

Blocklists have often been criticised for blocking legitimate email servers and being extremely

[3] An umbrella term for computer viruses, worms and trojans

slow to correct the error [Piquepaille, 2003]. A lack of any accountable standards body, such as ICANN that regulates DNS, has compounded the problem. Whilst the idea of Blocklists is a sound one, many see them as untrustworthy and over zealous. There are many alternatives available however.

2.3.1 Content Based Filtering

The actual content of the email itself can be analysed to determine if it is a legitimate message or not. In fact early spam filters using hand crafted rules - such as regular expressions - were quite effective [Rutgers, 2003]. Users of newsgroups and mailing lists often employed content filtering to classify mails according to keywords in the subject line or the senders address, the mails would then be sorted into folders based on this. When spam first began to appear on newsgroups and in email, content based filtering was the natural choice to combat it. Because spam emails often had characteristic words and phrases it was a simple matter to adapt existing rules to move spam into special folders or delete it entirely. But as spammers grew more sophisticated simple filtering using keywords became less effective.

This forced content based filtering to evolve, using machine learning (ML) techniques to automatically classify messages. Because of its proven text classification abilities [Lewis and Ringuette, 1994], the Naive Bayes method has become the focus of much research and development involving ML-based spam filtering. Naive Bayesian filters recognise emails that are similar to a training set of messages, over time the filter becomes more accurate at classifying messages. Naive Bayesian filtering has been implemented in client-side email applications such as Mozilla Thunderbird[4] and server-side filters like SpamAssasin[5].

2.3.2 Message Signatures

A message digest of a known spam email is created and published in a directory. Filters such as SpamAssasin can then query this directory and flag as spam any messages that hash to digests present in the directory. Since spam emails are often duplicated this has proven to be quite an effective technique.

The Razor[6] project implements this concept; users submit messages along with their one way hashes. Consistent successful reporting of known spam gives a user a higher rating of trustworthiness, meaning any spam they report in future will receive a higher priority for publishing in the directory.

2.3.3 Non-technical solutions

Non-technical solutions have mainly revolved around the formulation of new legislation or revising existing laws to make provisions for unsolicited bulk email. These measures have had little or no effect however, being confined to a single country or administrative region they fail to take into account the de-centralised nature of the Internet - a spammer or spam gang[7] can easily reside in one country and host their email servers in a country with less-stringent legislation.

The EU Directive on Privacy and Electronic Communications has attempted to make it illegal for any marketing information to be sent to an individual without their prior consent [EU, 2002]. Several member states, including Ireland, have adopted the legislation but the EU has been slow to take action against countries that failed to incorporate the directive into their own laws. After the deadline of 31st of October 2003, eight countries had not yet adopted the directive[8].

In the US, the most widely publicised piece of anti-spam legislation has been the Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003, or the CAN-SPAM act[9]. This act requires all marketing information sent

[4] http://www.mozilla.org/products/thunderbird/

[5] SpamAssasin is an extensible server-side filter - http://spamassassin.apache.org

[6] http://razor.sourceforge.net

[7] Spam operations consisting of a large number of professional spammers.

[8] Belgium, Germany, Greece, France, Luxembourg, the Netherlands, Portugal and Finland failed to pass the directive into law - http://www.insourced.com/article/articleview/1565/1/1/

[9] CAN-SPAM text available from the US Library of Congress - http://thomas.loc.gov/

by email to include legitimate return addresses and instructions on how to opt-out of the mailing list. But lawyers have claimed that the act cannot be enforced in a practical manner and, more seriously, that it supercedes stricter state laws that give members of the public the power to sue spammers [McCollum, 2004]. Because of its perceived leniency towards spammers, this act has often been referred to as the YOU-CAN-SPAM act.

International cooperation and common legislation appears to be the way forward for effective anti-spam laws. In July 2004 the USA, UK and Australia signed a "Memorandum of Understanding" that will allow governmental agencies in the three countries to share evidence against spammers and coordinate their enforcement efforts. The United Nations and the International Telecommunications Union have also indicated that they aim to standardise anti-spam legislation around the world in the next two years.

3 Spammers and Malware Authors

3.1 The Growing Connection : Attacks against blocklists

In November 2003 the Web Server hosting Spamhaus.org began receiving huge volumes of fabricated requests as part of a Distributed Denial of Service (DDoS) attack. The attack was launched from computers worldwide, that had been infected with the W32.Mimail.D virus.

This incident was just one in an increasing number of attacks launched using malware that infects machines with the purpose of using them as 'zombies' for sending spam or conducting DDoS attacks against anti-spam organisations.

Throughout August and September 2003, sustained DDoS attacks launched using malware caused at least three anti-spam systems to cease operations indefinitely[10]. The increasing sophistication of these attacks has highlighted a growing connection between spammers and malware authors.

3.2 Techniques and Tools

Today, spammers commonly employ 'Mass Mailer' worms to aid them in their activities. Mass Mailers are so called because they propagate by harvesting large amounts of email addresses from the target system to send copies of themselves to. Mass mailers are commonly designed to take advantage of Microsoft Outlook and Outlook Express, since these two email clients are in widespread use the worm will have more chance of success.

However worms have begun to emerge that have their own built-in SMTP engine, this allows the worm to send itself regardless of the email client being used, all that is required is TCP/IP port 25 to be accessible. The worm establishes a connection with an SMTP server (a remote server, or one that is part of the worm itself) that allows e-mails to be sent without verifying who is sending them or from where - this is possible because the SMTP protocol was designed long before the growth of the Internet, viruses and spam. As a result it is extremely permissive, allowing any information at all to be entered into header fields. Once established on target systems, spammers can use these worms for a wide range of activities; the most common being spam relaying, content hosting and denial of service attacks.

3.2.1 Spam Relays

Worms such as W32.SoBig.H, Migmaf and Fizzer install SMTP relay components onto the victim machine, allowing it to act as a proxy for large amounts of spam. In June 2004 the Network Management firm Sandvine determined that 80% of spam originated from zombie machines infected with trojans and worms [Sandvine, 2004], indicating an increasing tendency for spammers to use zombies as their preferred method of delivery.

[10] Monkeys.com, Compu.net and the SPEWS blocklist closed because of DDoS attacks

3.2.2 Content Hosting

The worm can have its own built-in HTTP server for hosting websites that the spammer advertises in their emails. Such content is often illegal so it is in the spammers best interest to host it in somewhere that allows them to remain anonymous and, if there are a large number of zombie machine involved, the website is almost impossible to shut down.

In the case of the Migmaf Trojan, the zombie machine acts as a reverse proxy for a master server hosting the actual content [LURHQ, 2003]. When a request is received for a web page, it is relayed to the master server through one of the infected machines. The master server then sends the page back along the same chain to the user that requested it. Thus, the spammer is able to host their content with possibly legitimate providers and effectively mask its true location.

3.2.3 Denial of Service (DoS) Attacks

In recent years, network attacks have been characterised by 'Denial of Service', or DoS attacks. This takes the form of flooding target computers and networks with traffic with the intention of degrading performance or disabling the system completely. DoS attacks can range from simple attacks originating on a single host to complex, distributed (DDoS) attacks that use multiple hosts and are much harder to trace.

The simplest type of DoS attack is a Ping flood. The Ping tool is useful for determining whether a system is properly connected to a network, and is available by default on most operating systems. It uses a form of data called Internet Control Message Protocol (ICMP) to send packets to a remote machine that sends a ping reply back acknowledging the request. Unfortunately, ping can also be used as part of a Denial of Service attack to 'flood' the intended target with multiple ping requests (ICMP packets) which cause the server to send back replies, resulting in network slowdowns and even crashes. A common technique is to spoof a source address for a large number of ping requests - this spoofed address is the target machine - the corresponding ping replies then overwhelm the target with no effect on the attacker.

Whilst ping floods using spoofed source addresses can be an effective means of bringing down a target system, there is still the possibility that the attacker can be traced since they must initiate the attack and send the ping request packets themselves. For this reason many DoS attacks are now carried out using ordinary home users machines infected with malware.

3.2.4 Bringing it all together : The Fizzer Worm

An extremely sophisticated example that provides all of the above 'features' can be found in the W32.HLLW.Fizzer worm which, along with W32.SoBig.H and W32.Mimail.E, was responsible for many of the attacks noted in section 3.1. Fizzer spreads by emailing copies of itself to contacts stored in Microsoft Outlook and Windows address books, and through the file sharing network Kazaa. Its payload consists of installing a web server for hosting the spammers content, an IRC (Internet Relay Chat) backdoor, an SMTP engine and DoS attack tools onto the victim machine. The worm then waits for instructions to be sent to it through the IRC backdoor. In this manner Fizzer can remain dormant and undetected on a victim machine, until it receives instructions to activate.

4 Distributed Blocklist

4.1 What is needed?

At its heart, a distributed blocklist is simply a system for the distribution of data, along the lines of the popular Freenet [Clarke et al., 2001], but with stricter controls over the integrity of the data. The data that is being distributed is a list of IP addresses for known sources (SMTP servers) of spam. In order to speed up queries, this data is stored according to the netblock of IP addresses it describes. It is important to define the features that would be desirable in a distributed blocklist.

- A trusted authority that controls the list, acting in a similar fashion to Certification Authorities for digital certificates.
- The list should be distributed over a commonly used protocol such as HTTP.
- It should be trivial for any entity to participate in the blocklist.
- Caching the results of queries locally improves efficiency by reducing repeated queries and moving data physically closer to where it most requested. This is commonly referred to as "Edge of Internet Caching", or cooperative caching [Lancellotti et al., 2002].
- The system should be resistant to poisoning attacks - corruption of the list by injecting false data. The trusted authority is key to this requirement.
- Resistance to DoS/DDoS attack. It should be extremely difficult to significantly affect or degrade this system. The trusted maintainers are easily visible targets and given sufficient resources an attacker may disable a large proportion of them, however the list would still exist and be accessible.

4.2 Design

The main activities that would be carried out by the entities in a distributed blocklist system would be *querying the list*, *joining the system* and *maintenance of the list*. Before these activities are outlined however, it is important to identify the entities that will participate in the distributed blocklist.

4.2.1 Entities

- Trusted Maintainers - Trustworthy entities that make decisions about what netblocks to list. The public keys of these entities should be distributed through a Public Key Infrastructure. These trusted maintainers may be blocklist operators that exist today, or well known organisations that already offer trust-based services such as Certification Authorities.
- Participants - Make up the backbone of the system by storing the blocklist. Each participant stores as little or as much of the blocklist as they want, but users of the system will also take part in it by caching the results of the queries they make.

4.2.2 Querying

The following simple algorithm details the steps that are taken to check if a specific IP is stored in the blocklist. For example, we wish to determine if 194.145.128.7 is listed, so we will access the section of the blocklist storing 194.145.*.* addresses.

1. Check the required data is not already in the cache.
2. If not, check another participant for an answer, this query may be referred until an answer is received (listed, or not listed).
3. Once an answer is received verify its signature using the maintainers public key and cache the answer.

4.2.3 Joining the System - Adding a new peer

- Peer A announces itself to maintainer server (by sending its location) and is given part of the IP space to store along with the location of another peer (Peer B) in the network and the maintainers public key. This data is digitally signed by the maintainer.
- Peer A then announces itself to Peer B. The message tells Peer B Peer A's location and what portion of the IP space it is storing. Peer B then adds this information to its 'routing table'. This message is retransmitted until a counter runs out.

4.2.4 Maintenance

Maintenance of the blocklist largely consists of determining what IP addresses to add to the list and in some cases, manual removal of IP addresses where removal has been sufficiently justified. The trusted maintainers will use their existing review processes when updating the blocklist. Current methods of updating non-distributed blocklists would still be applicable for the distributed blocklist, the only difference being the means for storing the blocklist data.

It is envisaged that this blocklist will aim to stop spam originating from ISPs that harbor spammers, or netblocks under the control of spammers. Sources such as ROKSO (Register of Known Spam Operations)[11] maintain an up to date listing of persons or organisations known to be involved in large-scale spamming. This information is useful in deciding what IPs and netblocks to list.

Spam originating from open proxies, such as machines infected with specifically designed malware, would be less straightforward to block since the IP addresses of these machines invariably change. This is due to the fact that the majority of infected machines are home users with ADSL or Cable internet connections where the IP address is allocated through DHCP.

[11] http://www.spamhaus.org/rokso/index.lasso

References

[Clarke et al., 2001] Clarke, I., Sandberg, O., Wiley, B., and Hong, T. W. (2001). Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46.

[EU, 2002] EU (2002). European union directive on privacy and electronic communications. Full text available from Eur-Lex, the portal to European Union Law at http://europa.eu.int/eur-lex/en/index.html.

[Lancellotti et al., 2002] Lancellotti, R., Ciciani, B., and Colajanni, M. (2002). A scalable architecture for cooperative web caching.

[Lewis and Ringuette, 1994] Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US.

[LURHQ, 2003] LURHQ (2003). Reverse-proxy spam trojan : Migmaf. Available from http://www.lurhq.com/migmaf.html.

[McCollum, 2004] McCollum, T. (2004). Usa tries to can spam. *ITAudit*. Article available at http://www.theiia.org/itaudit/index.cfm?fuseaction=foru

[Piquepaille, 2003] Piquepaille, R. (2003). Why blacklisting spammers is a bad idea. Available at http://radio.weblogs.com/0105910/categories/sidebars/20

[Radicati, 2003] Radicati (2003). Anti-spam market trends 2003-2007. Report available from http://www.radicati.com/.

[Rutgers, 2003] Rutgers, D. M. (2003). Statistics and the war on spam. *Statistics, A Guide to the Unknown.*

[Sandvine, 2004] Sandvine (2004). Trend analysis: Spam trojans and their impact on broadband service providers. Report available from http://www.sandvine.com/.

[Templeton, 2003] Templeton, B. (2003). Origin of the term ”spam” to mean net abuse - http://www.templetons.com/brad/spamterm.html.