2010

Clustering Spam Domains and Hosts: Anti-spam Forensics with Data Mining

Chun Wei
University of Alabama at Birmingham

CLUSTERING SPAM DOMAINS AND HOSTS: ANTI-SPAM FORENSICS WITH DATA MINING

by

CHUN WEI

ALAN P. SPRAGUE, COMMITTEE CHAIR
ANTHONY SKJELLUM
CHENGCUI ZHANG
KENT R. KERLEY
RANDAL VAUGHN

A DISSERTATION

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

BIRMINGHAM, ALABAMA

2010

CLUSTERING SPAM DOMAINS AND HOSTS: ANTI-SPAM FORENSICS WITH DATA MINING

CHUN WEI

COMPUTER AND INFORMATION SCIENCES

ABSTRACT

Spam related cyber crimes, including phishing, malware and online fraud, are a serious threat to society. Spam filtering has been the major weapon against spam for many years but failed to reduce the number of spam emails. To hinder spammers' capability of sending spam, their supporting infrastructure needs to be disrupted. Terminating spam hosts will greatly reduce spammers' profit and thwart their ability to commit spam-related cyber crimes. This research proposes an algorithm for clustering spam domains based on the hosting IP addresses and related email subjects. The algorithm can also detect significant hosts over a period of time. Experimental results show that when domain names are investigated, many seemingly unrelated spam emails are actually related. By using wildcard DNS records and constantly replacing old domains with new domains, spammers can effectively defeat URL or domain based blacklisting. Spammers also refresh hosting IP addresses occasionally, but less frequently than domains. The identified domains and their hosting IP addresses can be used by cyber-crime investigators as leads to trace the identities of spammers and shut down the related spamming infrastructure. This paper demonstrates how data mining can help to detect spam domains and their hosts for anti-spam forensic purposes.

Keywords: spam, forensics, clustering, data mining

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
C&C	Command & Control Server
CDN	Content Distribution Networks
DBL	Domain Block List
DNS	Domain Name Service
DOS	Denial-of-service
FFSN	Fast-Flux Service Networks
HMM	Hidden Markov Model
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	Identification
IP	Internet Protocol
IRC	Internet Relay Chat
ISP	Internet Service Provider
MIME	Multipurpose Internet Mail Extensions
OCR	Optical Character Recognition
P2P	Peer-to-Peer
RRDNS	Round-robin Domain Name Service
SPOF	Single-point-of-failure
SURBL	Spam URI Real-time Block List

SVM	Support Vector Machine
TLD	Top-level domain
TTL	Time to live
URI	Uniform Resource Identifier
URIBL	Uniform Resource Identifier Block List
URL	Uniform Resource Locator

1. INTRODUCTION

In recent years, due to its massive volume and spam-related cyber crimes, spam email has created a serious problem for society. According to the McAfee threat report last year (McAfee Avert Labs, 2009), there were 153 billion spam messages per day in 2008 and over 90% of emails were spam.

1.1 Current Spam Trend

Spam emails are no longer just unsolicited emails. Cyber criminals use spam to spread malware over the internet and infect other people's computers, to entice people to phishing sites that steal vital personal information, and to lure people into false transactions by exploiting human greed, such as promising lottery winnings, overseas inheritances, or easy work-at-home jobs with great salaries. Criminals also use spam to advertise counterfeit products and services, such as pharmaceuticals, luxury good, sexual-enhancement products and pirated software.

In 2008, a survey by the internet security company Marshal found that 29% of internet users had purchased products from spam because of the relatively cheaper price (M86 Security, 2008). The products, such as sexual-enhancement pills and luxurious watches, sold by spammers are counterfeit. But the buyers are willing to take the risk and purchase these products from spammers due to the competitive price. The revenues help the spammers to maintain their spamming network and to conduct various cyber crime

activities, such as online fraud, phishing and network intrusion, which lower the operation cost and make spamming a lucrative business.

1.2 Protective Mechanisms of Spammers

Common anti-spam techniques include spam filtering, URL and IP blacklisting. To avoid being detected, spammers are using a variety of methods to disguise their identities. To counter spam filtering, word obfuscation and image spam are used. To counter URL and IP blacklisting, botnets, multiple-IP hosting and Fast-Flux Service Networks (FFSN) are used. We will review these protective measures used by spammers.

1.2.1 Word Obfuscation

Because most spam filters are based on detecting keywords in spam, word obfuscation is used to obscure the keywords so that the filters cannot recognize them. Commonly seen obfuscation methods include deliberate misspelling, insertion of special characters, substitution by symbols and HTML redrawing. An article by Cockerham (2004) stated that there are over 6 hundred quintillion ways to spell the word "Viagra", while it is still recognizable by human eye. However, for a spam filter, it will be 6 hundred quintillion different words. Some obscured words can be reconstructed using computer programs, but others are beyond the capacity of Artificial Intelligence (AI). For example, the HTML redrawing can separate a keyword into letters and put each letter into a table cell. The letters can be colored, the cell can be formatted with colored background or borders. The html code will be too complicated for a spam filter to

determine what will eventually be displayed on the screen. Because there are so many variation obfuscation methods, it is almost impossible for a filter to recognize all of them.

Moreover, by using MIME, spammers can attach graphics to the email and have key messages embedded in the images, for example, the stock pump and dump scam. The Optical Character Recognition (OCR) techniques can be used to retrieve the text from the image. However, spammers can add noise to the image or distort the text to prevent the texts being successfully detected by OCR.

1.2.2 Botnet

A more effective way to stop spam is to block it at the source. If a mail server is detected as a spam sending machine, the IP address can be blocked and emails can no longer be sent. To avoid this single-point-of-failure (SPOF) scenario, more and more spam emails are sent by bots. A bot is a malware-infected computer, which will receive and execute commands from a command and control server (C&C) without the awareness of its legitimate user. In the first quarter of 2009, nearly twelve million new IP addresses were detected as bots, an increase of almost 50% from the last quarter of 2008 (McAfee Avert Labs, 2009). A group of bots that receive commands from the same C&C form a botnet. The botnets allow a spammer to send a large number of spam with little cost, 5 to 10 dollars per million spam messages (M86 Security, 2008) while not revealing the true location of the botmaster. About 80% of spam today can be accredited to fewer than 100 spam operations (Spamhaus ROKSO, 2010).

The botnets also make it difficult for spam investigators to track the origin of the spam emails because the sending IP addresses only lead to victimized computers. To

locate the C&C, an investigator has to further analyze the incoming and outgoing communication of bots, which may be massive. If a C&C is terminated, when the bots attempt to retrieve their next command, they find no command waiting and cease activity. A centralized C&C is still easy to detect and terminate. In order to protect the C&Cs, the notorious Storm Worm botnets adopted a distributed Peer-to-Peer (P2P) command structure (Grizzard, Sharma & Dagon, 2007). When a node is infected with Storm, it receives an initial list of possible "peer nodes" and attempts to contact each one to obtain a more current list of "peer nodes". This model has been more successful because of the Storm Worm's use of an existing P2P network, the Overnet network, to hide its traffic among the flow of traffic by as many as 1 million users who use the Overnet to illegally share music, movies, and software. The botnet structures will be reviewed in details in the next chapter.

Botnets are used to commit many cyber crimes, such as sending spam emails, launching denial-of-service (DOS) attacks and hosting spam websites. The shutdown of a botnet's C&C will greatly reduce the spam volume, for example, the decline in spam after the termination of rogue hosting provider McColo in late 2008 (Clayton, 2009; Mori, Esquivel, Akella, Shimoda & Goto, 2009). However, the spam volume bounced back within a month period of time (DiBenedetto, Massey, Papdopoulos & Walsh, 2009).

1.2.3 Spam Hosting Infrastructure

In a spammer's operation network, spam email is a means to an end. The spammer wants the email recipient to visit, usually a web link inside the emails. Figure 1 shows how a spammer operates his network to protect his identity and generate revenues.

The spammer controls the bots, infected computers, through a centralized C&C. He updates information on the C&C and from time to time each of his bots contacts the server to receive new commands, new spam templates and email address lists. Then the bots send out the spam emails with URLs pointing to spam websites. The spammer also maintains the websites on various web-hosts, as well as maintaining the corresponding DNS entries on name servers.

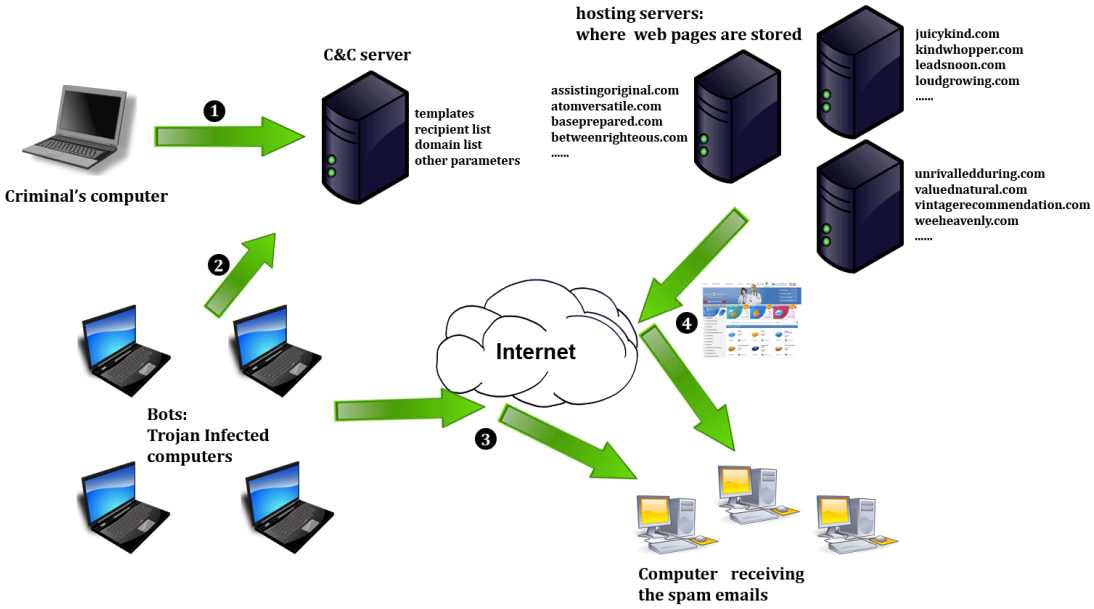

Figure 1: Information flow on a spamming network

From the above figure, we can see that the spam can be made ineffective if the hosting servers are taken down. If the users cannot reach the destination websites, no transaction will occur and the spammers cannot generate revenues. The same criteria applies to phishing and malware websites, no harm will be done if the websites are down.

Domain blacklisting is a common measure against spam domains, for example, SURBL/URIBL filtering (two popular spam "black lists" used by spam filtering

solutions). The URLs within the spam emails are analyzed and reported to the blacklist. Further incoming emails with blacklisted domains will be blocked. In order to protect the websites from block or termination, spammers combat domain blacklisting by registering a large number of new domains every day. Even though it costs more for spammers to register so many domains, St Sauver (2008) summarized several major benefits for spammers to do that: (1) to reduce the chance of spam being blocked by SURBL/URIBL filtering (two popular spam "black lists" used by spam filtering solutions) because new domains are less likely to be on the blacklist; (2) to reduce the risk of being prosecuted by law enforcement. Because the large volume of spam has been distributed among many different domain names, each will appear to be a small-volume spamming group, thus reducing the chance of catching law enforcement's attention; (3) to balance the traffic and increase the chance of survivability. In order to shut down the spam, one has to take down all of the domains or all of the back-end servers.

1.2.4 Fast-Flux Service Networks

Another emerging technology to protect the spam domains is Fast-Flux Service Networks (FFSN), which, as described by the Honeynet Project (2007), uses Round-Robin DNS (RRDNS) to disseminate the heavy traffic to a popular website to distributed machines as a way of load balancing. Upon a request, the DNS will use a round-robin algorithm to determine the IP address returned. By using botnets, a spammer can create a FFSN to serve a spam website. For each DNS lookup, the DNS server will return an IP address of a compromised computer. The compromised computer is usually a relay point. Through URL redirection or domain forwarding, a user is redirected to the real

hosting server where the web pages are located. Thus the IP address of the real server is protected.

The FFSN is a sophisticated technique that makes it harder to shut down the real website. However, our research showed that the majority of point-of-sale spam websites, such as pharmaceutical, luxury good and sexual-enhancement spam, are still using static IP for hosting and only use a large domain pool to combat domain blacklisting. FFSN is more frequently used in phishing and malware sites because the hosts for point-of-sale spam are still untouched by the anti-spam forces, while spam investigators eagerly pursue the phishing and malware spam.

1.3 Research Problem, Goal and Impact

Anti-spam research that tries to create better spam filters ignores the well-established concept of deterrence, "the inhibiting effect of sanctions on the criminal activity of people other than the sanctioned offender." (Blumstein, Cohen & Nagin, 1978, p.3). When society believes, and sees through repeated examples, that criminals are punished for their action, fewer people may become offenders. Spam filtering fails to deter spammers, as there is no real punishment. Every day billions of spam emails are filtered out, but most of them are either immediately discarded, or saved until a certain threshold of available storage is crossed, and then discarded, without ever being analyzed for their potential evidentiary value.

Spam can be more effectively stopped by disrupting its source, such as the C&C and hosting servers shown in Figure 1. This research targets the hosting servers because it is not necessary for the email recipient to find the origin of a spam email in order to

process the message, but it is essential that the spammer has an actual website where the consumer can buy his product. If the recipient cannot reach the sale website, no transaction can occur. The point-of-sale websites are where spammers generate most of their revenues. Researchers at University of California at San Diego (Kanich et al. 2008) studying the Storm Worm projected that the pharmaceutical spam portion of the Storm Worm activities may have generated as much as $350 Million for the botnet controllers.

This research develops a clustering algorithm to group spam domains that share approximately the same hosting infrastructure. The domain names that appear in the spam emails are clustered using the hosting IP addresses and associated email subjects. The email subject is used as additional evidence to group domain names whose hosting IP addresses partially match, but exhibit similarity in associated subjects.

The development of the clustering algorithm has gone through three stages. In the first stage, spam emails are clustered using a single-linkage algorithm, described in chapter 3: emails with identical attributes will be grouped. The email subject and domain name are used in experiment. The results have many false-positives because a common attribute may not necessarily mean two emails are related. There are cases when two emails share a common subject by chance. The results also have false-negatives because customized emails generated by templates have unique subject, even though they resemble each other.

In the second stage, a fuzzy string matching algorithm, described in chapter 4, is introduced to measure the degree of similarity between two strings. The algorithm can be applied to any email attribute that is a sequence of characters. In the experiment, the email subject was tested on the algorithm and produced promising results.

In the third stage, a derived attribute, hosting IP address, is combined with the email subject in clustering (described in chapter 5). The focus is also moved from spam emails to spam domains, which are closer to the spammer's end goal: to generate profit. The clustering of spam domains serves the same purpose as clustering emails because once a cluster of domains is confirmed, emails containing those domains can be easily retrieved. But the number of domain names is much less than the number of emails. The comparison at the individual email level is undesirable and unnecessarily, for example, there are many identical emails sent to different recipients. The clustering of domain names depends on an assumption that emails containing the same domains are related, which may not necessarily be true. Emails referring to a popular legitimate domain, such as Yahoo.com or Google.com, may not be related to each other at all, but emails referring to a domain created solely for spam purpose are usually related. Therefore, in the case of clustering spam domains, the assumption usually holds true.

The hosting IP addresses in leading clusters can be identified and used to trace the cluster over a period of time. If a cluster exhibits any significant patterns in the email subjects, the pattern may be used to check for future spam emails of the same genre. A cluster containing a large number of emails that cannot be matched to any historical cluster will be reported as an emerging spam campaign.

The identified hosting IP addresses can also be used to detect new spam domains hosted at the same location. If an IP address is notorious for hosting spam domains, new registered domains which resolve to the same IP address are likely to be spam domains as well. Whoever created these websites on the hosting servers is obviously responsible, either directly or as part of the same criminal conspiracy, for the spam emails that lead to

those websites. Our results showed that a small number of IP addresses are heavily used to host a large number of spam domains and remain active for a considerable period of time. Therefore, the hosting IP blacklist can improve the effectiveness of domain blacklist by detecting new spam domains without fetching the content of URLs. The DNS lookup will not encounter problems when fetching the web pages, for example, some of the hosting servers were found to deploy firewalls that block automatic probes.

The hosting IP address blacklist is also helpful for law enforcement personnel to target Internet Service Providers (ISPs) which provide bullet-proof service to spammers. Those ISPs will not question what is hosted there as long as the fees are paid.

The result of this research will improve the productivity of a spam investigator because it is beyond an individual investigator's capability to relate thousands of domains together through manual checking of WHOIS information or destination websites. Traditional law enforcement technology does not scale well in cases involving millions of data elements. The clustering algorithm will group spam domains that exhibit potential relationships. Then the spam investigators just have to check the validity of the clusters by sampling domains from the clusters and checking the WHOIS information and destination websites. If the destination websites cannot be reached, sample emails can be retrieved to check the validity.

This research explores a way of fighting spam at its source: the hosts of spam websites. If a source of the spam emails can be eliminated, the number of spam emails on the internet will be greatly reduced. The research will not only benefit spam filtering by improving domain and IP blacklisting, but also help spam investigators terminate the hosts of spam websites.

The fighting of spam is an ongoing process. A decade ago, spam filtering seems to be sufficient against spam. But spammers' techniques have so evolved that filtering itself is no longer adequate. Without any deterrence, spam volume has drastically increased over the past decade. We need to detect and terminate spam source so that spammers cannot send out spam so easily. We can also shutdown spam websites so that the spam messages become useless. Targeting the spam source has a deterrent effect and forces the spammers to invest in new technologies, and thus increase their operations cost. This should discourage more people from engaging spam-related cyber crimes. If they see more and more criminals being pursued and prosecuted, spammers will be less willing to take the risk inherent in criminal activity.

2. LITERATURE REVIEW

The goal of this spam research is to cluster spam emails and identify spamming infrastructure that belongs to the same spamming group. In this chapter, the related research is reviewed, including anti-spam and clustering algorithms on data streams.

2.1 Anti-Spam Research

When spam emails appeared about two decades ago, they were known as junk mail, or unsolicited commercial emails, and were unwanted by the recipients but were sent anyway by businesses to promote certain products. At that time, individuals used spam filters to filter out unsolicited emails. Therefore, in the beginning, the anti-spam research was focused on spam filtering techniques.

Spam is just a mean to an end, and that end soon began to include cyber crimes, such as phishing, malware and online fraud. Spam volume also rapidly grew after bots were used to send spam. Spam is not limited to emails, but now also appears in instant messaging, blogs and search engines. Just filtering spam at the recipient's end is no longer adequate. Therefore, the anti-spam research of filtering spam at the server level began to emerge, such as research on IP or URL blacklisting. Later, the research studying the spamming technologies and the cyber criminal activities began to appear, including the study of botnets, malware, spam destination websites and hosting infrastructure. There is also clustering and classification research on spam based on the email content or image attachments. Because there is a lot of spam research, in the

following sections, we will discuss only the most influential studies or those closely related to this research.

2.1.1 Spam Filtering

Early research on spam focused on building better spam filters, which distinguish spam from legitimate emails with high accuracy. The filter sets up a defense perimeter against unwanted email messages. Most spam filters rely on machine learning algorithms, which use training data to learn the rules that will predict future spam emails. Human-identified spam emails, along with legitimate emails are fed to the filters as training data. The assumption is that new spam will likely resemble the historical spam in some aspects. The filters can learn new trends by studying the false-positives and false negatives identified by humans.

Various machine learning techniques have been applied to spam filtering. Some of the most commonly used techniques are Bayesian approach (Sahami, Dumais, Heckerman & Horvitz, 1998), Support Vector Machines (SVM) (Drucker, Wu & Vapnik, 1999), Centroid-based approach (Soonthornphisaj, Chaikulseriwat & Piyanan, 2002), Neural Network (Clark, Koprinska & Poon, 2003), Genetic Algorithm (Sanpakdee, Walairacht & Walairacht, 2006), and Rough Set Theory (Zhao & Zhang 2005).

Because spam filters reply on word corpus to detect spam, they are susceptible to spam obfuscation. For example, we often see spam emails containing a big bag of words which has nothing to do with the email topic. The filter will probably regard the email as non-spam if those words are included in analysis. As the spam filtering techniques are improving, so are those of spammers. With MIME, spammers deploy various

obfuscation techniques to trick the filters, which will be discussed in the next section. Nevertheless, spam filtering is still the main weapon against spam today, even though fighting spam at its source has been viewed as a more effective way.

2.1.2 Message Obfuscation

Because of the limitation of spam filters, spammers can apply counter measures to trick filters to make the wrong decision. Some common obfuscation methods are: (1) substitution of certain character with similar symbols, for example, use "\/" to replace "V" in "Viagra"; (2) insertion of space or special characters between letters, for example, "Vi-a-gra" for "Viagra"; (3) purposeful misspelling of a word, for example, "Bacheelor" for "Bachelor" ; (4) insertion of irrelevant paragraphs; (5) text-embedded images and (6) HTML redrawing. Spammers usually will combine several obfuscation techniques together to outwit the filters.

Figure 2 shows a spam email that used HTML redrawing and insertion of irrelevant paragraphs. The highlighted text was arranged in HTML table cells. The letters are carefully spaced to make it readable to human eyes, which is "BRAND NEW VIAGRA AND CIALIS". The paragraph at the bottom is just plain text, but it has nothing to do with the email topic. The email subject did not reveal the content of the email either. The colored letters also serve as noise in the background. Therefore, it is very hard for a filter to figure out what the message is trying to say. As a result, a filter that replies on spam word corpus will fail to recognize it as spam.

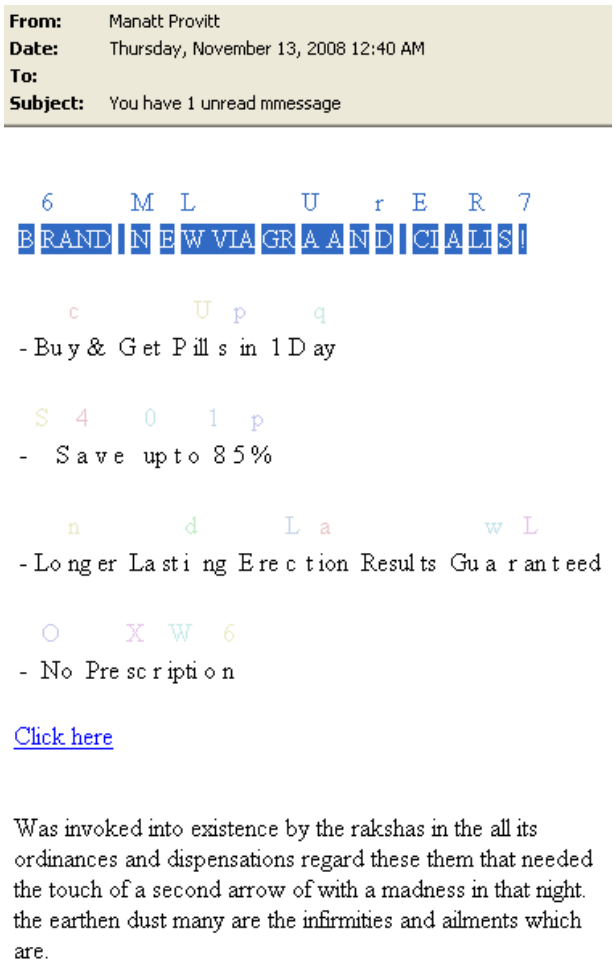

Figure 2: An obfuscated spam email using HTML redrawing

Lee and Ng (2005) used lexicon tree Hidden Markov Model (HMM) to remove obfuscation from spam emails and recover the true messages, which can then be checked by regular spam filters. The model proved to be useful against four word obfuscation methods: misspellings, insertions, substitution and segmentation. Lee, Jeong & Choi (2007) improved their model to a dynamically-weighted HMM which have shorter runtime. Liu and Stamm (2007) described a de-obfuscation method that dealt with the Unicode letter transliteration, the substitution of an English letter with a Unicode character. Bergholz et al. (2008) proposed a method that could recover text from HTML

emails. However, their experiment was limited to two tricks: font size and color, which are primitive obfuscation techniques seen in today's spam. The techniques used by spammers keep evolving and the above research is not adequate.

Text embedded graphics are also used by spammers to defeat word-based spam filters. As an answer, research on image spam began to draw attention. The research by Zhang, Chen, Chen, Yang and Warner (2009) used OCR technique to extract the text from images and clustered emails based on image similarity. Other image spam classification research include near-duplicate detection (Wang, Josephson, Lv, Charikar & Li, 2007), content obscuring detection (Biggio, Fumera, Pillai & Roli, 2007), extracted overlay text and color features (Aradhey, Gregory & James, 2005), a maximal figure-of-merit learning algorithm (Byun, Lee, Webb & Pu, 2007), and a combined framework for text-and-image spam (Byun, Lee, Webb, Irani & Pu, 2009). On the other hand, spammers are also improving their techniques. Figure 3 illustrated an image spam with word and image obfuscation. Note the text in the image is wave-shape, which is intended to prevent OCR from detecting the letters.

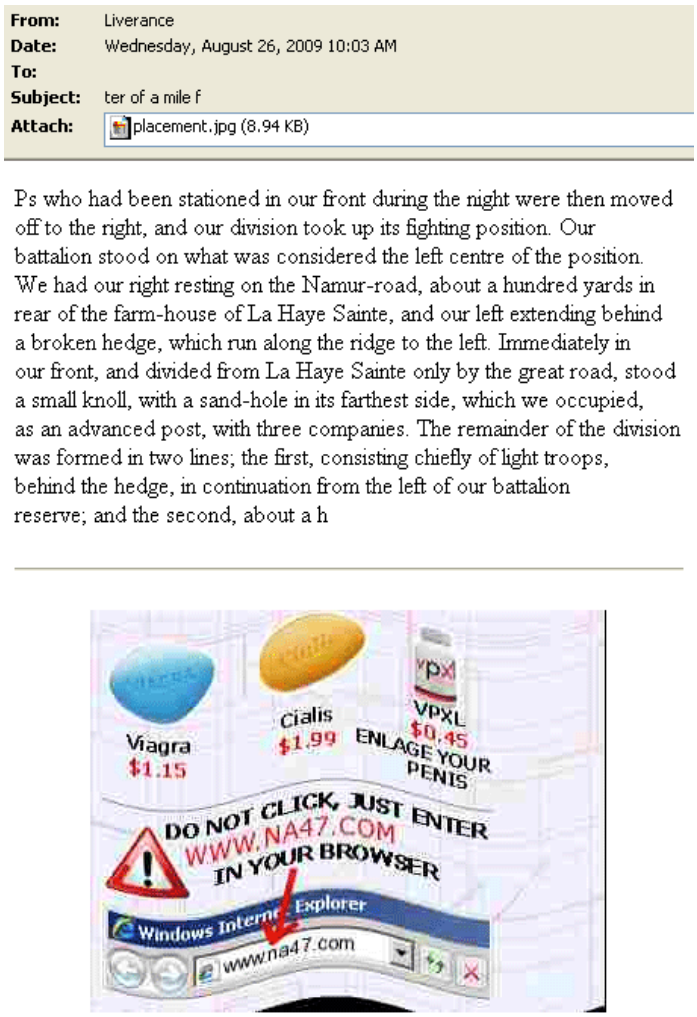

Figure 3: A spam email with distorted text in an image

2.1.3 Research on Botnet Detection

With the growth of spam volume and the use of botnets, it is more desirable to filter spam at network level. The spam filtering at the client side does not stop the spam from being sent out. Even though the emails do not reach their recipients, they go through the internet traffic and waste considerable energy and resources. Therefore, a better way to terminate spam is to block them at the source. Most spam nowadays is

sent by botnets, a group of infected computers, which inspired much research on botnet detection.

Ramachandran and Feamster (2006) studied the network-level behaviors of spammers and found the majority of spam was sent from a few concentrated portions of IP address space. However, the top 3 networks on their list were Korean Internet Exchange, China Telecom and Sprint. Obviously, we cannot just block all emails from Sprint or China Telecom. Because botnets are used to send spam emails, Ramachandran, Feamster and Dagon (2006) tried to detect possible bots by observing the lookups to Domain Name System-based Blackhole Lists (DNSBLs), which are lists of IP addresses that originate spam. The same group (Ramachandran, Dagon & Feamster, 2006) did a preliminary study on the effectiveness of DNSBLs and found only 5% of bot IP addresses were ever listed at Spamhaus Policy Block List (Spamhaus PBL, 2010). Ramachandran, Feamster and Vempala (2007) tried to detect botnets by analyzing the behavioral patterns of sending machines because a bot controlled by malware will send a batch of emails at a fixed time interval while a normal user won't exhibit that pattern. But the detection of botnets is still a passive countermeasure against spam. The prevention of computers from infection and disruption of the command and control servers (C&Cs) would be more effective against botnets than blocking the sending IP addresses (Cooke, Jahanian & McPherson, 2005).

Gu, Zhang and Lee (2008) proposed an abnormality-based detection system, BotSniffer, to identify centralized botnet C&Cs. Bots controlled by the same C&C will exhibit spatial and temporal correlation and similarity during their malicious activities, such as DNS attacks, propagation and online fraud. Based on the correlation, the botnet

members can be detected and the C&C be traced. However, the system is not effective against P2P C&C structure used by the Storm Worm (Grizzard et al. 2007; Holz, Steiner, Dahl, Biersack & Freiling, 2008). In a centralized C&C structure, each bot logs into the same Internet Relay Chat (IRC) channel and communicate with the C&C server. If the server or the channel is taken down, the whole botnet becomes ineffective. But in P2P C&C structure, each bot uses HTTP-based protocol to communicate with other bots, which is stealthier than IRC because the communication between bots can be hidden in normal Web traffic. Figure 4 shows the difference between a centralized C&C and a P2P C&C. Some commonly used P2P protocols include Gnutella (Kirk, 2003), Chord (Stoica, Morris, Karger, Kaashoek & Balakrishnan, 2001) and Kademlia (Maymounkov and Mazières, 2002). The P2P protocols were initially used for internet users to share music and videos, (in some cases, this involves copyright violation). Now spammers are using P2P C&C infrastructure to protect the botnets. Each bot can play the role of a client or a server. Using the Storm worm as an example, the botmaster publishes commanding files over the P2P network through some super-nodes (also known as seeds). A newly joined bot uses an initial list of contact and will contact listed peers to retrieve a more recent list of peers (may include the seeds), from which it retrieves hash tables to locate the commanding files. Later, the bot can be turned into a server and serve other newly joined bots. Therefore, shutting down a P2P botnets is very difficult because the system is so distributed that taking down some nodes will virtually cause no damage to the botnets. Gu, Perdisci, Zhang and Lee (2008) proposed BotMiner, which is independent from the botnet protocol and structure, to detect botnet membership. Their assumption is that bots controlled by the same botmaster will exhibit similar patterns in their

communications and malicious activities. The BotMiner will study the log files and find correlations. However, the paper did not discuss how to locate the botmaster once the botnet is detected. Other interesting botnet research includes Rishi, a system for detecting IRC-based botnets by using known nickname patterns (Goebel & Holz, 2007), clustering and classification of network flow traffic based on IRC-like traffic patterns (Strayer, Walsh, Livadas & Lapsley, 2006; Karasaridis, Rexroad & Hoeflin, 2007) and BotHunter (Gu, Porras, Yegneswaran, Fong & Lee, 2007), which detects and monitors bots' behaviors by following a malware infection life cycle dialog model.

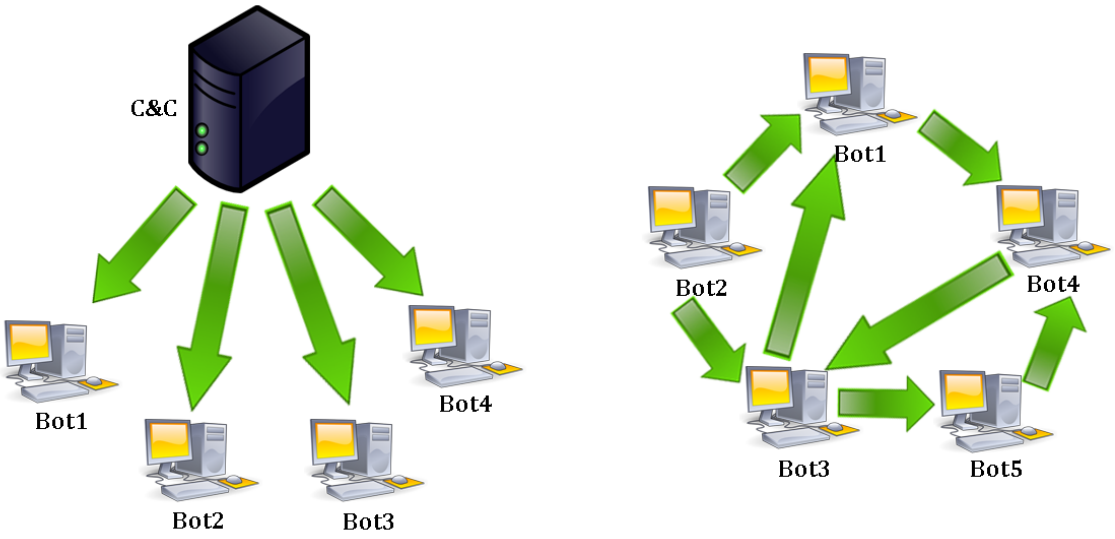

Figure 4: Botnet structures: (Left) centralized C&C; (Right) Peer-to-Peer

Some P2P botnets still involve super-nodes, which will expedite the search for commands (Schoof & Koning, 2007). The recent shutdown of Waledac botnet (Claburn, 2010) was the result of targeting the domains which provide instructions for Waledac bots. About 227 domains were taken offline when the nameservers were terminated

(Kaplan, 2010). These domains are used by newly-recruited bot to look up the C&C servers. Even though the bots are still out there, they are ineffective because they cannot find the instructions without the DNS service. Therefore, the clustering of spam domains is still useful against botnets. If a set of domains are found to be used by botnets, we can find the nameservers and take them down.

2.1.4 Research on URLs and Spam Hosts

Most spam filters use email content to decide if an email is spam. The content of a spam email can tell a lot about the spammer, for example, the URLs in the email. The URLs point to websites where the vital actions take place for spammers to make a profit. A spammer can forge sending email address and sender's name because it is not necessary for the email recipient to be able to find the true origin of a spam email in order to process the message. However, the URLs are usually real because it is essential to the delivery of the spammer's end goal, the sale of a product or service, for an actual location of the advertised website to be reachable to the email recipient. If the recipient cannot reach the point-of-sale website, no transaction can occur and the spam email becomes useless. The same applies to a phishing website, if the user cannot open the website, no harm will be done. Therefore, detecting the IP addresses of spam domain names appearing in spam messages will be another effective way of stopping spam by minimizing its revenue. If a hosting IP is shut down, all related domain names will be ineffective before they can be moved to a new host. New spam domains can be easily detected by checking the hosting IP addresses if they are still alive.

Several spam researches have studied the URLs in spam messages. Calais et al. (2008) used four attributes (language, message type, message layout and URLs) to cluster spam campaigns. In their paper, a spam campaign is a group of messages that have the same goal and use the same obfuscation strategy. Emails collected by honeypots in several Brazilian networks were grouped based on common frequent features. Some big spam groups they reported actually consist of more than 100,000 spam messages. The paper also investigated the network patterns of the sending machines (abuse of HTTP, SOCKS proxies and open relays). The paper did not further investigate the URLs, such as fetching the web pages, finding hosting IP addresses or WHOIS information.

Pu and Webb (2006) observed trends in spam email message construction, especially obfuscation methods in HTML-based spam emails. They then built a Webb Spam Corpus, which consists of nearly 350,000 web pages that are obtained from URLs in the HTML-based spam emails (Webb, Caverlee & Pu, 2006). They also found that the web hosts in their Corpus were tightly connected to each other by web links. But the graph was too heavily clustered to see any detailed information of how the hosts were actually connected. Using the Webb Spam Corpus, they categorized the web pages into five categories: Ad Farms, Parked Domains, Advertisements, Pornography and Redirection (Webb, Caverlee & Pu, 2007). They found web spam pages tend to have more duplicates and redirections than normal web pages. They also identified the 10 hosting IP addresses with the most web page count and found that two IP ranges accounted for 84% of the hosting IP addresses. However, they did not indicate whether these IP addresses were related or not. Later, they used hosting IP address as a feature in clustering web spam because they found the hosting IP range of spam hosts fairly

differed from the IP range of legitimate hosts (Webb, Caverlee & Pu, 2008). However, on the IP range over 204.*, the distinction was not so clear.

The Spamscatter project (Anderson, Fleizach, Savage & Voelker, 2007) also fetched web pages using the links in spam emails and clustered the web pages based on screen shot similarity. They categorized scams based on the content of the websites. Although they did not formally define the term "scam", it can be inferred from the paper that a scam is a group of related websites that promote the same product or service. The ten largest virtual-hosted scam categories they listed contained three "watches" categories, two "pharmacy" categories and two "software" categories but there was no indication whether they were related or not. They traced domains for about two weeks and found that multiple virtual hosts (different domains served by the same server) and multiple physical hosts (different IP addresses) are infrequent. This may no longer be true because spammers are improving their hosting infrastructures to protect the servers. For example, the largest cluster found in our research contains many domains that point to the same website. The Spamscatter project also investigated the lifetime of scam hosts and found the majority of them were short-lived. However, a spammer can point a website to a different IP address by changing DNS entries and creating new domain names to replace old ones that are blacklisted. Therefore, the termination of a host or domain name does not necessarily mean a scam has ended. In our study, the largest cluster lasts for the entire experiment period, while new domain names are introduced every day and hosting IP addresses are shifted from time to time.

By using the botnets and RRDNS lookup, spammers recently created FFSN to host spam domains. A domain served by FFSN can point to many IP addresses, and each

of them has a short time to live (TTL) value. At one time, the IP lookup of the domain will return IP address A. Several minutes later, it will return IP address B and so forth. Therefore, a domain served by FFSN is difficult to shut down given the number of IP addresses.

Zdrnja, Brownlee and Wessels (2007) studied DNS response data to detect hostnames served by FFSN, which would be associated with many A records. Passerini, Paleari, Martignoni and Bruschi (2008) developed FluXOR, a system that can distinguish hostnames that use FFSN from benign hosts that use RRDNS and distributed mirror servers and monitor the FFSN to find the botnet membership. Holz, Corecki, Rieck and Freiling (2008) described in detail how the FFSN operates, the detection of a FFSN served domain from normal domains served by Content Distribution Networks (CDN) and possible mitigation strategies. Konte, Feamster and Jung (2009) studied the point-of-sale spam domains hosted using FFSN. They collected about 3000 domain names from over 115,000 emails in 2007 and found many point-of-sale domains were hosted at distributed machines with each IP address serving for a short period of time before replaced by a new IP. However, our research found that FFSN is more often used to host phishing and malware spam websites than point-of-sale websites.

Despite the large number of hosting IP addresses, a single-flux network can be brought down by targeting the nameservers, which are still static. A double-flux network is even more sophisticated because the nameservers are also fast-flux. However, a double-flux network is harder to maintain than a single-flux network.

2.1.5 Scam vs. Spam Campaign

A scam is a group of related websites that promote the same product or service according to the Spamscatter paper (Anderson et al, 2007). Examples of scams include ED pills scam, E-card scam, pump and dump scam. A scam usually lasts longer than a spam campaign, which is a group of messages that have the same goal and use the same obfuscation strategy (Calais et al, 2008). Just like a brand can run many advertising campaigns, a scam can run several spam campaigns. Email messages belonging to the same campaign may resemble each other, indicating that they may originate from the same botnet which uses a template to generate spam. In our research, we will use the two terms consistent with the previous research.

2.2 Research on Data Clustering

The goal of this research is to group spam emails based on common attributes. In this section, the research on clustering will be reviewed.

2.2.1 Linkage Based Clustering

The most intuitive clustering algorithm is the single-linkage clustering algorithm, also known as nearest neighbor clustering, which is implemented by SLINK (Sibson, 1973). Starting from a random data point, the algorithm finds the nearest neighbors of that data point (the distance is often defined by dissimilarity coefficient). Then the algorithm recursively finds the nearest neighbors of the newly added data points found in the previous iteration until all data points are visited. Tom (2008) used a method similar to the single-linkage algorithm. Spam emails were grouped if any of the following three

attributes were identical: sending IP address, message body and email subject. The biggest cluster reported contained 85% of all emails in a 9-day period in Dec 2007. The emails were related to replica watches, gambling, porn and sexual enhancement.

The linkage between emails is subject to several errors. Considering only identical subjects and message bodies will not find email messages with customized subjects or message bodies that are generated by templates. The emails will look similar except for the customized username or email address. Moreover, emails with common subjects like "Re:", "Fwd:" and "Urgent" are often seen in spam messages. Emails with these subjects are not necessarily related to each other. Therefore, the result will likely to contain both false-positives and false negatives.

Another challenge is to define the distance (or dissimilarity) between emails. Each mail is associated with multiple attributes that may suggest connections between them. In Tom's paper, the distance is either zero (at least one identical attribute) or infinite (no identical attribute). It would be more desirable to measure the similarity between two emails using a scale instead of the binary dichotomy and each attribute will contribute to the overall similarity score. The more common attributes two emails share, the higher the similarity score is. Depending on the empirical evidence some attributes may have more weight than others.

Jeh and Widom (2002) proposed SimRank to calculate the average similarity between two objects that are associated by other entities. In their experiment, the objects are scientific papers. The similarity between two papers depends on the number of common words in their titles and the number of common papers that are referenced by both papers. In the case of spam emails, it would be common email attributes, such as

subject, sender's IP or referred URL. To find the similarity between two emails, we need to find the similarity of all the associated attributes. However, the email attributes are not exactly like the attributes in scientific papers. For example, the title of a paper always indicates the topic of the paper, while an email subject may say nothing about the email content. In this research, we decided to cluster spam domains instead of emails because we wanted to target the hosting places of spam domains and there were too many near-identical emails sent to different recipients. We developed algorithms to compute the similarity for email subjects and the IP addresses of the URLs.

2.2.2 Connected Components

After the similarity score is computed, a graph can be constructed to link related spam domains. Each domain is a vertex in the graph and the similarity score is the edge weight. A threshold can be set to decide whether an edge can be dropped. Then the cluster can be retrieved by searching for connected components.

In an undirected graph *G*, two vertices *u* and *v* are called connected if a path can be found from *u* to *v*. A graph is called connected if every pair of distinct vertices in the graph is connected. A connected component is a maximal connected subgraph of *G*. Each vertex belongs to exactly one connected component, as does each edge (Bollobas, 1998).

There are several criteria to choosing a connected component. The simplest way is to use the single-linkage: there exists at least one path between each pair of vertices in the component. However, this measure tends to be weak because there may be two groups of vertices which are only connected by a single vertex (Figure 5) or a single edge.

The vertex might contain some common attributes shared by both groups, such as a common email subject or URL link. In this case, the cluster is somewhat questionable.

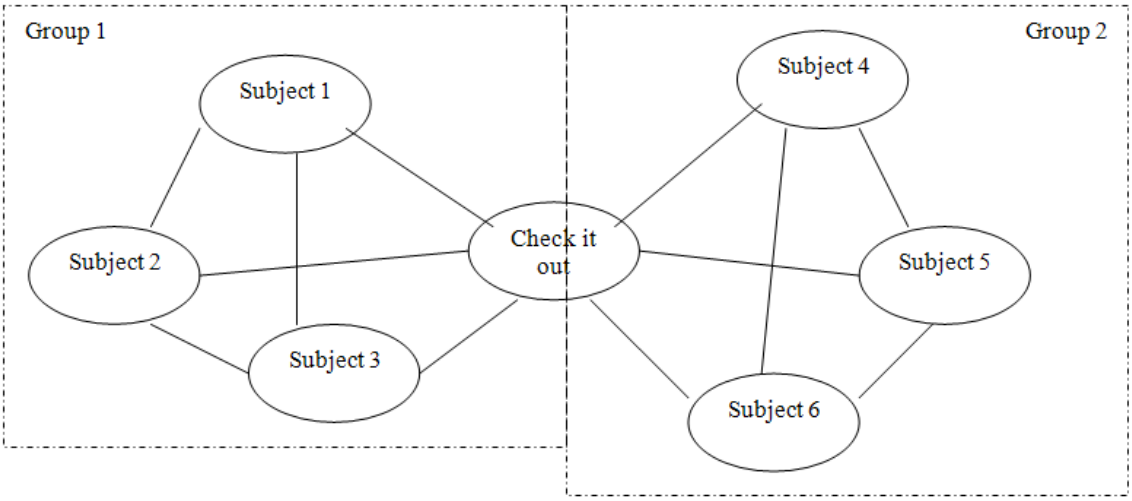

Figure 5: False clustering caused by an ambiguous subject

The most rigorous check is the use of Clique (Bollobas, 1998), which is a fully-connected graph: each pair of vertices is connected by an edge. Groups 1 and 2 in Figure 5 are cliques. However, this check might be too strong. We may get many small clusters while more valuable evidence may show they are related.

A moderate approach is to use the bi-connected component. A connected graph is bi-connected if and only if it contains no articulation point, also called a cut vertex (Baase, 1988). The removal of an articulation point will cause the graph to be disconnected. Therefore, between each pair of vertices in a bi-connected component, there exist at least two different paths, meaning no common vertices along the paths. This check will separate the two groups illustrated in the figure.

A more complex graph connectivity theory, the vertex cut or the edge cut (Bollobas, 1998), can be used, which is stronger than the bi-connected component. A cut or vertex cut of a connected graph G is a set W of vertices whose removal renders G-W disconnected. The vertex connectivity $\kappa(G)$ is the size of a smallest vertex cut that separates G . A graph is called k-vertex-connected if its vertex connectivity is k or greater. A vertex cut for two vertices u and v is a set of vertices whose removal from the graph separates u and v. The local connectivity $\kappa(u,v)$ is the size of a smallest vertex cut separating u and v. Local connectivity is symmetric for undirected graphs, so $\kappa(u,v)=\kappa(v,u)$. Moreover, $\kappa(G)$ equals the minimum of $\kappa(u,v)$ over all pairs of vertices u,v.

The edge connectivity $\lambda(G)$ is defined analogously when vertices are replaced by edges. Thus an edge cut of graph G is a set of edges whose removal renders the G disconnected. The edge-connectivity $\lambda(G)$ is the size of a smallest edge cut, and the local edge-connectivity $\lambda(u,v)$ of two vertices u,v is the size of a smallest edge cut disconnecting u from v. Again, local edge-connectivity is symmetric. A graph is called k-edge-connected if its edge connectivity is k or greater.

In the case of spam domain clustering, we can find the minimum cut of a cluster. The higher the cut value, the more strongly connected is the cluster. A threshold can be used to decide whether a cluster is strong enough to be accepted. However, the minimum cut algorithm is more complex than the bi-connected component. In our experiment, the bi-connected component appears to produce satisfactory results; therefore the graph connectivity algorithm is not implemented.

2.2.3 Research on Data Streams

The research on data streams is mentioned here because spam emails have some aspects common to data streams: (1) extreme huge volume, it is estimated more than 153 billion spam per day; in our dataset, the daily spam emails can reach 1 million per day; (2) new spam emails every minute , the spam emails keep coming, the clustering of emails is an ongoing process; (3) evolving patterns: new obfuscation techniques to evade spam filters, new campaigns (new phishing and malware), new URLs and spam domains.

Since new spam emails appear each day, it is not efficient to rebuild the clusters if new data arrives. We developed an algorithm that can group new emails as they arrive and link new clusters to old ones. The design is motivated by research in data streams. The nature of data streams demands three critical requirements for clustering algorithms (Barbara, 2002): (1) Compressed representation of data; (2) Fast, incremental processing of newly arriving data points; (3) Identification of outliers. In data stream research, data compression is achieved by Clustering Feature (CF) used in the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), a clustering method for very large databases (Zhang, Ramakrishnan & Livny, 1996). Instead of storing all the data points, a CF stores three critical attributes that represent the cluster: the number of data points, the linear sum and the square sum of all the points in a cluster.

CF = (N, $\sum_{i=1}^{N} X_i$, $\sum_{i=1}^{N} X_i^2$) X_i is a data point, which is a vector or scalar.

The data summaries stored in CF vectors are sufficient to calculate the statistics required for clustering, such as centroid, radius and diameter for a cluster using the following formula:

Centroid $\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N}$

Radius $R = \sqrt{\frac{\sum_{i=1}^{N} (X_i - \overline{X})^2}{N}}$

Diameter $D = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (X_i - X_j)^2}{N(N-1)}}$

Subsequent research papers (Aggarwal, Han, Wang & Yu, 2003; Cao, Ester, Qian & Zhou, 2006; Zhou, Cao, Qian & Jin, 2007) developed similar data stream clustering algorithms based on CF tree. The data stream algorithms have been applied to server logs to detect network intrusions. Spam data is different from server logs because logs contain numeric attributes or nominal attributes that can be transformed into binary attributes. The transformation is accomplished by creating a new attribute for each possible value of the original attribute and use '1' or '0' to interpret the presence or the absence of that value. Emails contain nominal attributes that cannot be transformed into binary representation because the infinite number of possible values, for example, the number of different email subjects cannot be exhausted. Therefore, we cannot define an email using a vector representation. As a result, it is impossible to define a centroid or radius for a spam cluster using coordinates, which are used to define micro-clusters in data stream papers.

However, the data stream papers present an interesting concept of clustering data points incrementally. Aggarwal et al. (2003) added the sum of time stamps and the sum of squares of time squares into the CF, which can be used to calculate the mean and the standard deviation of the arrival times of data points in a cluster. The time stamp statistics are used to estimate the recency of a cluster.

Cao et al. (2006) introduced a decay function based on the time stamps of data points and a cluster can be weighted using the decay function. A cluster with more recent data points will have a higher weight than a cluster with more historical data points.

In our research, we would like to trace spam domains and their hosting IP addresses throughout their life time. The hosting IP addresses can relate spam domains that serve the same scam. A historical domain may still be interesting if new domains are pointing to the same IP location. A historical IP is also useful because it can be compared with new IP to identify a network that is patronized by a spammer. The spam domains can be clustered based on a time interval, daily or hourly, and then be compared to historical clusters to find if there is any domain name, IP address or email subject in common. If a match is found, the historical cluster can be updated. If not, the cluster will be reported as an emerging threat. Because we are interested in leading clusters with a great number of emails and domain names, a small cluster will be treated as an outlier and ignored.

3. HIERARCHICAL CLUSTERING

The first experiment on clustering spam emails used an agglomerative hierarchical clustering method combined with connected components (Wei, Sprague, Warner & Skjellum. 2008). The goal was to group spam emails with common attributes. This chapter will describe the data preparation, algorithms, experimental results, and how they lead to the changes proposed in the following chapters.

3.1 Attribute Extraction

Before clustering emails, useful attributes need to be extracted from emails. In our research, the email parser extracts eight attributes from the email messages: sender's IP address, sender's email, email subject, email body length, email word count, URLs, attachment filename and attachment size. Another attribute message ID is assigned to each message as an index. Some attributes are broken down into two sub-attributes, for example, the URL may be broken into a hostname and a path. Some attributes may not be present in all emails, such as the URLs. Therefore a separate table is created for URLs, which is linked to the main table by the message ID. Likewise for the email attachment, an attachment table is created.

Apart from the inherent attributes that can be directly parsed from the email (the eight attributes just mentioned), derived attributes, those that cannot be directly acquired from emails but can be derived from inherent attributes by looking them up in additional sources, are also important, for example, the WHOIS data from the Domain Name

Registrar and fetched web pages of URLs. The derived attributes provide further evidence of any relationship between the spam emails and spammers. For example, if two different URLs point to the same website, then they are related; and if two spam domains are hosted at the same IP address, then the two domains are related. Derived attributes are useful in finding non-obvious relationships and validating initial clusters built from inherent attributes. In the first experiment, only inherent attributes, the domain name portion of the URL and the email subject, are used for clustering. Derived attributes, such as the destination website and the WHOIS information are used for validation. In a later experiment described in Chapter 5, the derived attribute, hosting IP address, is used for clustering.

3.2 Clustering Methods

Two clustering methods have been used in this experiment. The agglomerative hierarchical algorithm is used for grouping emails with common attributes. After this clustering method is applied, the largest cluster contains several different types of spam emails, indicating there is a problem of false-positives, emails that are not related are grouped. Next, the connected component with weighted edges algorithm is used to mitigate this false positive situation. If a cluster resulting from the first method is found to be weak, the second algorithm is applied to the cluster to break down it into smaller clusters of higher quality.

3.2.1 Agglomerative Hierarchical Clustering Based on Common Attributes

An agglomerative clustering method is used for global clustering in order to group spam emails based on common values of email attributes. In the beginning, each email message by itself is a single cluster. Then clusters that share a common attribute are merged. Each time a new attribute is introduced, clusters from the previous iteration will be merged based on the common values in the new attribute. The old clustering results are backed up in case the process needs to be reversed due to false positives.

D(i, j) is defined as the distance between cluster i and j. D(i, j) = 0 if cluster i and j share a common value in an attribute and D(i, j) = ∞ if not. Two clusters are merged if distance is 0. A common attribute value means exact string matching.

In our experiment, the email subject is used in the first iteration of global clustering. Therefore, two clusters are merged if they share a common subject. The email subject is used because almost all emails have a subject and, in most cases, two emails with the same subject are identical. There are exceptions, such as subjects that are blank or contain common phrases, which cause false-positives in the result.

The domain name portion of the URL is used as the attribute for the second iteration. A domain name is the part of a URL that is the human readable representation of an IP address. A DNS lookup is used to return the IP address of a domain name. Two clusters are merged if they contain emails which point to the same domain.

Figure 6 is an example of how spam emails are merged using subject and domain names. At the first level, emails are grouped into three clusters based on sharing an identical subject. In the next level, Email 4 in the cluster “No need to feel shy” shares a common domain “ddffy.com” with the emails in the cluster “Enhance you manliness.”

Email 5 shares a common domain “ddvood.com” with emails in the cluster “Get more pleasure in love.” As a result, all three clusters are merged into one cluster after the domain iteration.

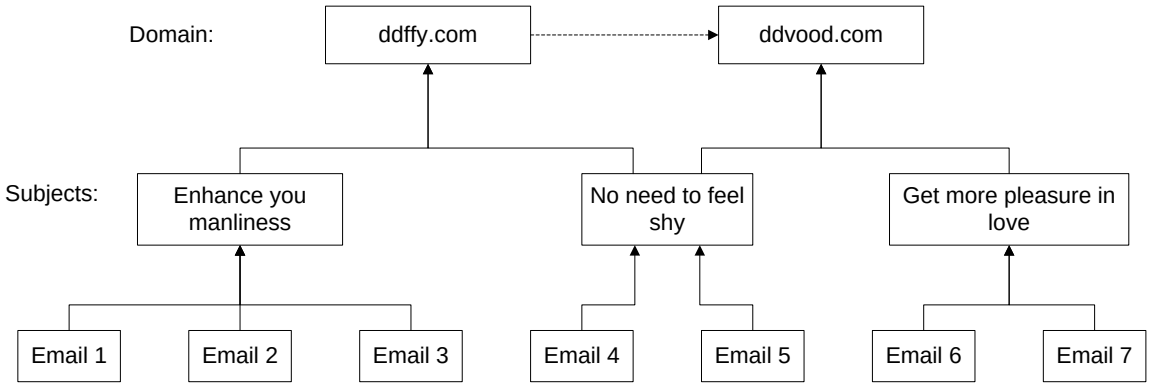

Figure 6: Merge clusters based on common subjects and domains

The agglomerative clustering method is desirable because the number of clusters decrease as the clustering process iterates. In the second iteration, only domain names of a cluster with same subject are compared, instead of comparing the domain names in individual emails. The weakness of the method is that coincidence, common phrases and sheer luck can cause untrustworthy relationships to be introduced since our logic is that two emails are linked as long as they share at least one common attribute. In our experiment, we stopped after two iterations because we have encountered a false-positive problem: the biggest cluster contains more than 67% of the emails with URLs. To counter false-positives, a connected component using the weighted edge method is introduced in the next section to break the biggest cluster into smaller clusters.

3.2.2 Connected Components with Weighted Edges

To reduce the error of grouping unrelated scams into the same cluster, the concept of "connected component of weighted edges" was applied.

A *connected component* (Baase, 1988) in an (undirected) graph is a set *S* of vertices such that for every vertex *v* of *S*, the set of vertices reachable (by paths) from *v* is precisely *S*. The weight of an edge shows the strength of the connection between the two vertices. The goal is to find connected components of this graph, considering only edges with weight above a threshold. This goal follows this reasoning: Suppose a spammer owns 10 domains and has a list of 10 subjects, and he sends out emails by randomly picking a subject and a domain. There are 100 possible combinations. If he sends out enough emails and we have collected enough emails, we should see examples of all 100 combinations. So if domains are assigned as vertices and subjects as edges, we will find that the ten domains are tightly connected to each other with strong edges. Yet, if two domains owned by two different spammers are connected to each other by chance because the two spammers share a common subject, the connection between domains will be weak since the probability of two spammers picking the same subject is relatively lower. If a group of domains in the biggest cluster are tightly connected to each other, they are likely to be owned by the same spammer.

Therefore, all domains from the biggest cluster are retrieved and assigned as vertices. The edges connecting them will be any common subject and the weight of the edge is the number of common subjects shared by two domains. A threshold is then selected and all edges with weight below that threshold will be dropped. The remaining connected components should be tightly related.

The algorithm is designed to allow the threshold to be adjusted to produce a more favorable result. By applying the algorithm to a cluster that has false positives, the cluster is divided into smaller clusters that are more tightly related. If the results still show too many false positives in our sub-clusters, the threshold will be incremented. Or if the results show too many tiny clusters, the threshold will be decremented. In the experiment, thresholds 2, 3 and 5 are used and the results turned out to be most accurate with threshold 3, which will be explained in more detail in section 3.3.4.

3.3 Experimental Results

This section describes the data set used in the experiment and the results of two methods described in section 3.2.1 and 3.2.2.

3.3.1 Data Collection

The dataset consists of three months of email submitted by a volunteer who has manually identified the messages as spam. This volunteer collects a high volume of spam through the use of "catch all" domain configuration. A "catch all" configuration accepts emails for all possible users, even non-existent ones, at a given domain. One common technique spammers use to "harvest" new target email addresses is to send emails to randomly generated usernames at the targeted domains. Emails which do not bounce back are assumed by the spammer to have been delivered, which implies that the username does exist. Because of the "catch all" address configuration, all spam sent to that domain is accepted, thus attracting more spam emails in the future. The "catch all"

domain contributed 211,000 spam emails during the months of June, July and August of 2007.

3.3.2 Results of Agglomerative Hierarchical Clustering

In the beginning, each of the 211,000 messages of our dataset is a cluster by itself. In the first iteration, emails with the exact subject are brought together, so that the number of clusters became equal to the number of subjects, which is 72,160, with the largest cluster containing 9,380 emails sharing a common subject.

The next attribute used for clustering is the domain portion of the URLs found within the bodies of the spam email messages. A cluster containing emails with no URL will not be affected by the second iteration and remains intact. There are 33,993 out of 72,160 clusters that contain emails with URLs, which will be included for the second iteration.

After the second iteration of the algorithm, clustering by Subject x Domain, the number of clusters is reduced from 33,993 clusters to 3,247 clusters. Each of the newly formed clusters was formed by linking existing clusters that share at least one common domain. 89.7% of all of the email fell into 42 clusters when this process was applied. Many smaller clusters existed as well, but our focus is on the larger clusters, as our goal is to identify the greatest nexus of criminal spamming activity.

3.3.3 Validation of Results

Using a visual inspection method, the resulting clusters were evaluated manually. Because the clusters in this experiment were conjoined by a common “domain” portion

of their URL, a routine was developed to fetch and save a graphical image, or thumbnail, of each destination website's home page. Where the resulting collection of website images from a single cluster were visually inspected and determined to be the same, a high confidence was placed upon the integrity of the cluster. Where the resulting collection of websites contained divergent images, a second level of validity checking was required.

For the second level validity checking, a list of the internet domains contained in a given cluster is checked by using a "WHOIS" command which returns information about the hosting IP addresses, registrar information and the nameservers of the domains.

First Level Validation: Website Image Comparison. The first level validity checking reports the majority of the top clusters, except for the largest one and a few others, as being "highly trustworthy" based on the identical or nearly identical images which are returned when the corresponding web pages are retrieved. The largest seven clusters are listed in Table 1. These seven clusters are out of a total of 42 clusters containing more than 100 messages each.

The domains in clusters B, C, D, F, and G point to the same website, indicating that all of the spam messages in each of these clusters are advertising the same product or service. Therefore, these clusters are of high-validity.

Table 1: Top 7 clusters from June to August, 2007

Cluster	Number of emails	Number of subjects	Number of Domains	Scams
A	105,848	16,125	10,845	Various
B	3,810	112	20	Downloadable Software
C	1,284	48	37	Elite Herbal
D	851	13	62	Downloadable Software
E	744	224	157	Various
F	584	125	207	ED Pill Store
G	554	88	9	Diamond Replicas

Second Level Validation: WHOIS Data. Domains in Cluster A and E point to different websites, indicating that messages in the two clusters were being used for different scams. Secondary validation, WHOIS information lookup, is required to determine if these messages were indeed related. The WHOIS information of a domain contains the hosting IP addresses, the nameservers and the contact information of the person who registered the domain. If two domains are hosted at the same IP address or served by the same nameserver, or were registered by the same person, they are likely to be related. Spam investigators can use these commonalities to determine if the relationship was strong enough. For instance, two domains using the nameserver "ns1.yahoo.com" is weak, but the relationship between two domains using the nameserver "ns1.strawpusnips.com" is strong since that nameserver appears to be solely for spam domains.

In Cluster E, 22 unique websites were found among 100 domains. However, the WHOIS data showed that the 100 domains were hosted at only 6 IP addresses, linked to only 2 sets of owner registration data and 3 sets of nameservers. Further investigation found all 22 websites on each of the 6 IP addresses, and on domains registered by either of the two owners, and on domains served by all of the three nameservers. Therefore, Cluster E proved to be a valid cluster even though it contains 22 different scams.

Primary image validation of Cluster A revealed many destination websites, including "Canadian Pharmacy", "ED Pill Store", "Elite Herbal", "Herbal King", "International Legal RX", "My Canadian Pharmacy", "Penis Enhancement Patch", and "US Drugs". Other websites only contain a small number of domains, some of which were identified as non-spam websites.

Cluster A was subjected to 2nd level validation. The returned WHOIS information was also divergent, indicating Cluster A was questionable and should be divided into smaller clusters.

3.3.4 Results of Weighted Edges

To markedly reduce the chance conjoining of unrelated scams, the concept of "Weighted Edges" was applied. With this model, comparisons are made between the vertices and their edges, but rather than a single common attribute sufficient to force the merging of two clusters. Each common attribute will increment the edge weight towards a threshold value. The algorithm is designed to allow the threshold to be adjusted, and the validation processes are repeated to determine if the new threshold delivers a more favorable result. Beginning with a cluster which has failed to show strong

trustworthiness, the weighted edges algorithm is applied. Edges with weight below threshold value are dropped and associated vertices disconnected. Remaining connected components form sub-clusters.

Through experimentation with threshold values of 2, 3, 5, it was determined that for our current email population, 3 was an ideal threshold value for achieving the most trustworthy sub-clusters.

After the "Weighted Edges" algorithm was applied to Cluster A, with a population of 10,845 domains, 26 big sub-clusters along with many tiny-size clusters and singletons (size 1) were formed.

The 26 big sub-clusters were then validated using the methods described in 3.3.3. Websites in sub-clusters 2 through 26 showed a very high correlation visually, indicating each of the sub-clusters was strongly related.

However, sub-cluster 1, after applying "Weighted Edges", still had a number of distinct visual patterns present in destination websites. Some of these distinct patterns, such as "Herbal King" and "Elite Herbal" were proved to be related by the WHOIS validation. 125 associated domains were registered by a guy named "Danny Lee" from "Health Worldwide, Inc" in Kowloon, Hong Kong. All domains that were still alive among the 125 domains were served by the nameserver "ns1.chongdns99.com", and hosted on the IP address "210.14.128.34". 65 additional domains were registered by "Sammy Lee" from "Liquid Ventures, Inc", also in Kowloon, Hong Kong. Among the 65 domains inspected, the active domains used the same nameserver and hosting IP as the ones in the first group. However, the secondary validation found other domains in sub-cluster 1 to be unrelated to each other.

3.4 Discussion

The agglomerative clustering method used a single-linkage approach to cluster spam emails. During each iteration, clusters from previous iteration are grouped if they share common attributes. The algorithm suffers a problem that emails may share a common attribute as a result of coincidence. Figure 7 shows an example how sexual enhancement spam can be merged with replica watch spam by a common subject "Satisfaction guaranteed," which appears in both spam. Except for that particular subject, there are no other subjects or domain names in common. Therefore, the linkage is susceptible and probably should be negated.

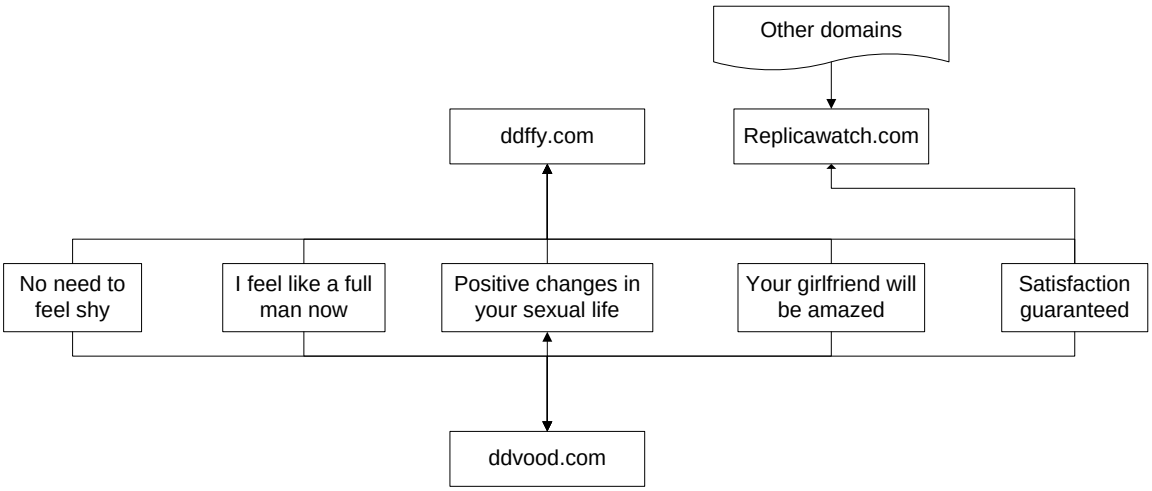

Figure 7: Accidental linkage by a common subject

Due to occurrences of accidental linkage, the resulting clusters need to be validated by human after each iteration. A "connected-component with weighted edges" algorithm is used if the validation finds that a cluster contains false-positives.

However, to validate and re-cluster at each iteration is not efficient. The validation still requires a lot of human power and many small-size clusters can be grouped if more attributes are used.

On one hand, identical email subject may not necessarily prove two emails are related. On the other hand, customized email subjects resembling each other are strong evidence that they are crafted by a single spammer who uses templates to generate a unique subject for each recipient. In chapter 4, a fuzzy matching algorithm for email subjects will be introduced.

Spam domains can also relate to each other in a similar pattern if they are registered by a spammer in a batch. The DNS lookup may show that they are served by the same host and nameserver. Therefore, they can be grouped without even fetching the destination web pages. In chapter 5, hosting IP address, a derived attribute from domain name, is included in the clustering.

4. FUZZY STRING MATCHING

When spammers use malware to send spam emails, they can create templates and malware can automatically fill in the keywords to generate customized spam messages. For example, we often see email subjects which read: "Special 80% discount for customer username on all Pfizer." In this case, the username can be replaced by a real email address and "80%" can be replaced by another discount amount. Therefore, each email will be somewhat unique. From time to time, spammers can make small changes to the template and create a variation of the old pattern. For example, delete "all" from the above subject and add "product" after "Pfizer." In order to detect and group customized spam emails generated using templates, a fuzzy string matching algorithm (Wei, Sprague & Warner, 2009) is introduced here.

4.1 String Similarity

To measure the similarity between two strings, we need to find the portion of strings that matches. The Levenshtein distance and dynamic programming is applied to find the optimal alignment of characters between two strings.

4.1.1 Inverse Levenshtein Distance

The most common way to measure disagreement between strings is through edit distance, also referred as Levenshtein distance (Levenshtein, 1966), which has been used extensively in approximate string matching by using the bottom-up dynamic

programming algorithm (Gusfield, 1997). Because we want to measure the similarity rather than distance, we use dynamic programming to find the alignment between a pair of strings *s* and *t* that maximizes the number of matches. The resulting number of matches between strings *s* and *t* is called their inverse Levenshtein distance, written as ILD(*s*,*t*). For example,

String *s*: r e l a t i o n _

String *t*: r o t a t i - n g

There are five matching letters, therefore ILD(*s*, *t*) = 5.

4.1.2 String Similarity

The measure of string similarity between a pair of strings expresses the portion of the strings that match. The measure is preferred to be always between 0 and 1. The Kulczynski coefficient accomplishes this but is defined for sets instead of strings. The Kulczynski coefficient on sets A and B is defined by:

Kulczynski (A, B) = (|A∩B|/|A| + |A∩B|/|B|) / 2

where |A| and |B| are the size of set A and B, |A∩B| is the size of the intersection. It yields a value between 0 and 1.

A Kulczynski coefficient for strings is defined in a way analogous to sets. Having the number of matches from the alignment, the Kulczynski coefficient for strings s and t is defined by:

Kulczynski (*s*,*t*) = (ILD(*s*,*t*)/|*s*| + ILD(*s*,*t*)/|*t*|) / 2

where |*s*| and |*t*| are the length of strings *s* and *t*.

Therefore Kulczynski(“relation”, “rotating”) = (5/8 + 5/8) / 2 = 0.625.

There are other coefficients that can be used for computing similarity. The Simpson coefficient allows a smaller sub-set match to a bigger set. The Jaccard coefficient favors toward sets with equal sizes. In our research, we do not want to favor either case, therefore, we decided to choose the Kulczynski coefficient, which takes the average of two sets.

4.2 Subject Similarity

Email subject similarity can be measured in the same way as string similarity. A subject may contain multiple tokens. A *token* is defined as a sequence of nonblank characters in a subject; tokens are separated by spaces. A subject will be regarded as a sequence (or string) of tokens. The number of tokens will be defined as the subject length, analogous to the string length as the number of characters in the string.

4.2.1 Subject Similarity Score Based on Partial Token Matching

Since a subject is a string of tokens, we can compute similarity of subjects analogous to string similarity: the similarity of subjects *a* and *b* is computed as Kulczynski(*a*, *b*), *a* and *b* are matched as two strings, where each token in *a* and *b* is treated like a character in a string.

However, each token is actually a string of characters. We observed some tokens that could partially match each other because they were generated by a pattern to produce variation in email subjects. For example, look at the discount amount in the following two subjects:

February 70% OFF February 75% OFF

Therefore, when matching a pair of tokens, we allow tokens to partially match each other if they have the same length. In particular, if two tokens *p* and *q* have the same number of characters, say *n* characters: length*(p)* = length*(q)* = *n*, we define match(*p*, *q*) = *m*/*n* where *m* is the number of matching characters. The matching is done like this: for each character $(p_1, p_2, ..., p_n)$ in *p* and $(q_1, q_2, ..., q_n)$ in *q*, compare p_i with q_i. Hence match(*p*, *p*) = 1. Thus the matching score for the above example is 2.667/3 = 0.89, because 70% is partially matched to 71%, yielding a score of 0.667.

4.2.2 Adjustable Similarity Score Based on Subject Length

Some subjects are longer than others, containing more tokens. The chance of two long subjects matching each another is much less than that of two short subjects matching each other, while yielding approximately the same similarity score.

Consider the following two groups of subjects:

Group 1:

3a06c0.c15a38's discount #VUUkNK. BEEST Quaelity MedDs.

3a2061bf.5640c7's discount #MeEhEi. BEEST Quaelity MedDs.

Group 2:

RE: Discount Sale

Discount Sale

Both will yield a similarity score of 0.67. But the relationship of subjects in the 2nd group is obviously weaker than the first one. Therefore, a coefficient is introduced to adjust the subject similarity score based on the subject length. The purpose of the coefficient is to decrease the credit given to short subjects that match each other.

According to the statistics of our dataset, about 60% of all subjects have 5 or fewer tokens. We consider 5 to be the critical length: if the average subject length of two subjects being compared is 5 or more, the coefficient will be 1, but if their average subject length is less than 5, the coefficient will be less than 1, decreasing the credit for matching. The similarity score for subjects *a* and *b* will be:

*Similarity(a,b) = C * Kulczynski*(a,*b*),

where $C = \sqrt{\min(\frac{|a|+|b|}{2 \times MaxLength},1)} = \sqrt{\min(\frac{|a|+|b|}{10},1)}$

4.3 Subject Clustering Algorithms

This section describes a clustering algorithm for grouping similar email subjects based on the fuzzy string matching algorithm in previous sections. A pattern matching problem is: given a pattern P, find in a set of strings $S=\{S_1, S_2, ..., S_n\}$ all strings matching to P. The clustering algorithm will find in S all interesting patterns and the corresponding strings. We describe 2 versions of clustering algorithm, with the second one building on the first one.

4.3.1 Simple Algorithm

The first algorithm is called the simple algorithm. Let S be the set of subjects, and let S_0 = S. To form one group (one cluster): Select an arbitrary subject *s* in S_0, usually the first one. Then let Group(*s*) = {*t*: Similarity (*s*,*t*) $\geq$ *h*} (where *h* is a similarity threshold) and remove Group(*s*) from S_0. This results in a new group being formed, and S_0 has shrunk. This process is repeated until S_0 is empty. Now the set S of subjects have

been partitioned into groups, some large and some small. However, this strategy is order dependent. Depending on which subject we pick as the seed, the result will be different. Take the following three spam subjects for example:

100mg x 90 pills $159.95 buy now

$159.95 100mg x 90 pills buy now

$159.95 Viagra 100mg x 30 pills price

The first subject is similar to the second, and the second is similar to the third. But the difference between the first and the last may be too big to be considered as similar. Therefore, if the second subject is chosen as the seed initially, all three subjects will be put into one group. But if the first or the last is selected as the seed, two groups might be seen in the result.

4.3.2 Recursive Seed Selection Algorithm

To improve the first simple algorithm, a second algorithm, the recursive seed selection algorithm, is used. This algorithm is like the simple algorithm, but after a group Group(s) is formed, we attempt to enlarge it by selecting an s' in Group(s) (but s' relatively far from s), and adjoin all subjects t such that match(s',t)$\geq h$. Subject s' is picked from Group(s) where Similarity(s', s) is minimum. New s' is repeatedly selected until no new t can be drawn to the group.

The pseudocode for the recursive seed selection strategy is in Appendix B.

4.4 Experimental Results

The experimental data consists of spam emails sampled from the month of May, June and July of 2008. The first day for each month is picked. The spam emails were contributed by a volunteer, who has a "catch all" configuration for his domains, which is explained in 3.3.1. Starting from May 2008, there are approximately over 10,000 up to 20,000 spam messages and 3000 to 6000 distinct email subjects every day. Note there is a 1:3 ratio between the number of subjects and the number of email messages (Table 2).

Table 2: Email and subject count

	May 1st 2008	Jun 1st 2008	Jul 1st 2008
Email count	13258	12370	16669
Subject count	4396	3568	5095

Experimental results showed the recursive clustering algorithm out-performed the simple algorithm when there were variations in the pattern. The simple algorithm regarded each variation as a separate cluster.

There are still questionable clusters. For example, there is similarity among the following subjects. However, just looking at the subject does not provide sufficient evidence to reach a decision. Further evidence, such as the email content or referred URLs, is required.

Be not afraid of making changes in their lives

Do not afraid to make changes in their lives

Do not limited in their desires

Do not want to their stores?

Do not limit himself to your wishes

Do not restricted to your desires

Do not want to buy unknown them in stores?

No limit to their wishes

It is not restricted to your desires

No limit himself to your wishes

There are also variations in patterns that are not picked up by the algorithm when the subjects are short. For example,

Medications Discount for 193659710.85053065891344

Medications Discount for 3a06c610.bf6ddcd8

and

Meds Coupon for 344481546.24852211963912

Meds Coupon for 357132225.97966732638608

When the subject similarity score is in the grey area, very close to the threshold, it is necessary to use additional attributes. If two emails point to the same websites, the relationship is strengthened. In the following chapter, the hosting IP address will be added as a derived attribute to the clustering algorithm.

5. CLUSTERING SPAM DOMAINS

This chapter describes the method used to cluster spam domain names based on hosting IP addresses and email subjects (Wei, Sprague, Warner, Skjellum, 2010). The idea is motivated by the fact that spammers use many domain names to minimize the damage caused by domain name blacklisting and increase site availability. The domain names often follow a pattern, indicating that they are created by an automated process. The WHOIS information will reveal that these domain names are correlated. Once the domain names are clustered, the emails referring those domain names are considered to be originated from the same spam organization.

5.1 Retrieval of Spam Domain Data

Attributes that can be directly extracted from an email header and content are called inherent attributes. Extracted inherent attributes include email subject, sender's name, sender's email address, sender's IP address, date received, embedded URLs, and email attachment. Among them the URL is the most interesting, because it leads to the hosting spam websites. In our dataset, over 90% of spam emails contain URLs in the email text. Some spam use image attachment and have URLs embedded in the image, therefore we are actively working to incorporate OCR into our system in order to detect the URLs in the image. In this paper, only the emails with URLs in text format are included.

Derived attributes are information derived from inherent attributes. The URL can be used to fetch the websites, the hosting IP addresses and WHOIS information. Derived attributes provide more useful information leading to the spam origin than inherent attributes. On the other hand, some URLs may point to websites which are no longer available, such as the Ad Farms and Parked Domain pages found by Webb et al. (2007). Therefore, the information is harder to retrieve. In this research, the hosting IP address of the domain name and associated email subject are retrieved (Figure 8). The retrieval of hosting IP requires three steps: extracting URLs from emails; extracting the domain name portion of the URL; fetching the IP address of the domain name.

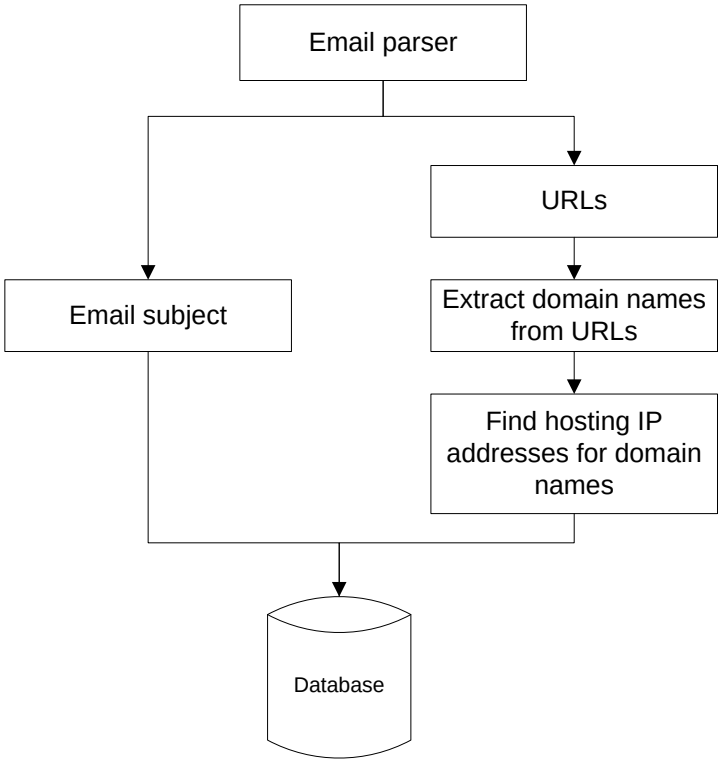

Figure 8: Retrieval of clustering attributes

5.1.1 Wildcard DNS Record

During the process of extracting the domain name portion of the URL, we observed that many spam domains use wildcard DNS records. By using this technique, random hostnames can be generated from a registered domain.

A wildcard DNS record is a DNS record that will resolve requests for non-existent hostnames with a matched domain suffix (Wikipedia, 2009). It is specified by using a "*" as the left most part of a hostname, e.g. "*.domain.com." Therefore, if a user requests a hostname ending with "domain.com" that does not have a corresponding entry in the DNS records, the wildcard record can resolve the request. The wildcard DNS record helps to resolve a hostname when a user only knows the domain name and is uncertain about the hostname. Spammers are taking advantage of this to combat URL blacklisting which includes the whole hostname.

If a spam domain uses wildcard DNS, fetching the WHOIS information of that domain once will be sufficient. To test a wildcard domain, we first extract the domain name portion from the host name, e.g. the domain name for "zhpt.tarecahol.cn" would be "tarecahol.cn". Then we create our own phantom host name by attaching a random string to the domain name. If the new host name can still be resolved, and provides the same data as the original, it strongly indicates the domain is using wildcard DNS records. Then it is very likely that all other host names ending with the same domain name resolve to the same site. This strategy greatly reduces the number of hostnames that need to be fetched.

5.1.2 Retrieval of Hosting IP Addresses

The UNIX "dig +short [hostname]" command is used to check the IP address of the advertised hostname. Since a domain can be hosted on more than one IP address and an IP address can host many domains, there is a many-to-many relationship between domain and IP. Each domain-IP pair is saved as a unique entry in the database table (Appendix A). We also record the date when the domain is first observed in spam emails and the last time it is observed. The WHOIS information for each IP is also retrieved using the "dig" command, and we store the network block, organization name, country code and ASN number in another table. The two tables are linked by IP index.

5.2 Daily Clustering Methods

Since new emails are added to the database every day, an on-going clustering method for spam emails is desirable. It is not effective to re-cluster the entire data set each time we receive new emails. We want to cluster new emails as soon as they arrive and identify relationships between the new clusters and the previous clusters. Therefore, a daily clustering strategy is developed and then we link clusters in two adjacent days if they share similar email attributes. In doing so, we can find what the clusters look like in the recent days as well as tracing them back to find out what they look like historically.

The daily clustering algorithm categorizes spam domains into different groups based on where they are hosted and the subject line of the associated emails. Sometimes the subject line indicates the actual content of the emails, sometimes it does not. But even when the subject has nothing to do with the content of the email, similar subjects resembling a pattern are strong evidence for showing a relationship among the emails.

Figure 9 shows the clustering algorithm procedure. First, the subject similarity and IP similarity are computed for each pair of domain names. Second, the domain pairs with similarity score exceeding a threshold are linked. Third, a bi-connected component algorithm is used to group related spam domains. Last, clusters with a large number of emails are exported.

To relate two domain names, we measure the similarity between their host IP addresses and their associated email subjects. The similarity of email subjects has already been defined in chapter 4. In this chapter, we will define the similarity between two IP addresses. Because each domain may correspond to multiple IP addresses and are associated with many email subjects, similarity coefficients will be used to compute similarity between two sets of IP addresses and email subjects.

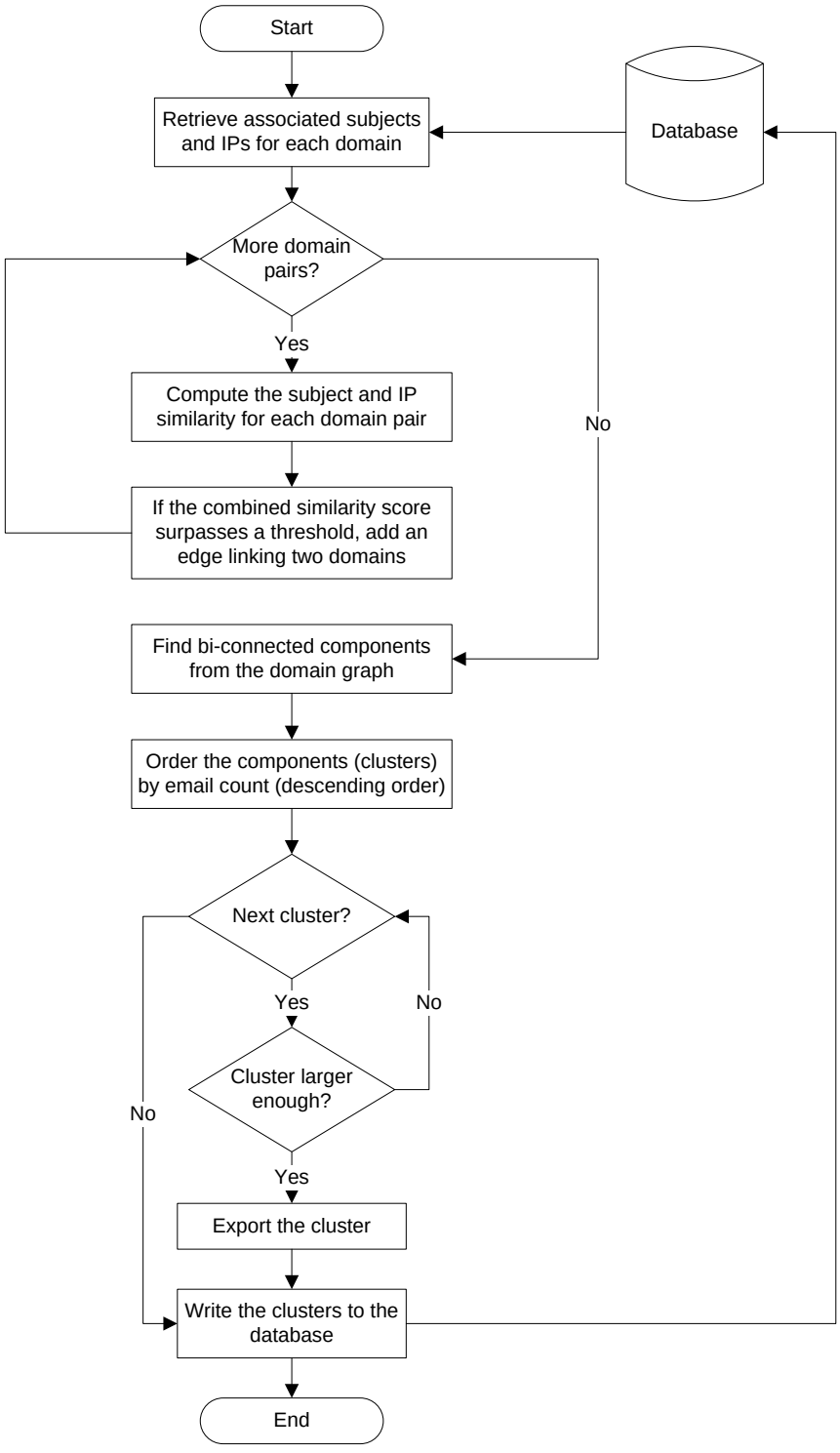

Figure 9: Daily clustering algorithm

5.2.1 Hosting IP Similarity between Two Domains

A domain name can be resolved to several IP addresses as a way of load balancing and improving search results. The traffic to a website can be distributed among several IP addresses. The DNS server will direct requests to different IP addresses based on the order they arrive. If the domain has three IP entries, usually the *n*th request will go to (n%3)th IP address. Sometimes, the locale of the IP is considered as the DNS tries to point the user to the nearest server. A user residing in the United States will be directed to a server in US, while a user in China will be directed to a server in an Asian region. These tactics enable spammers to increase their site availability and to make it difficult to trace the server location. Therefore, the comparison of IP addresses between two domain names becomes a set operation. The Kulczynski coefficient is used to measure the similarity between two IP address sets.

The Kulczynski coefficient on sets A and B is defined by:

Kulczynski (A, B) = (|A∩B|/|A| + |A∩B|/|B|) / 2, where |A| and |B| are the size of set A and B. It yields a value between 0 and 1.

When matching two IP addresses, a little fuzziness is allowed. Two IP addresses can be partially matched if they belong to the same subnet, which is recognized by matching the first three octets. For example, 1.2.3.4 partially matches 1.2.3.5 and a score of 0.5 is assigned.

For IP sets A and B, |A| <= |B|, we match each IP address in A to all IP addresses in B and choose the maximum matching score S_i. The sum of S_i is $\sum_1^n S_i$ (|A| = n, |A| <=|B|), which replaces the |A∩B| in the Kulczynski coefficient formula.

Some domains have many hosting IP addresses, some domains have fewer. In considering the IP set size, the chance of two sets of size four matching each another is much less than two sets of size one matching each other. If a pair of domains each corresponds to four IP addresses and matches perfectly, it is unlikely that this occurs by chance. Therefore, based on the size of an IP set, a coefficient is added to adjust the IP similarity score.

According to our dataset statistics, only 10% of all domains resolve to more than 4 IP addresses. Therefore, the maximum size is set to 4. If the average size of two IP sets being compared is larger than 4, the coefficient is set to 1.

The IP similarity score will be:

*S(A,B) = C * Kulczynski*(A, B),

where $C = \sqrt{\min(\frac{|A|+|B|}{2\times MaxSize},1)} = \sqrt{\min(\frac{|A|+|B|}{8},1)}$

For example, if domain A has IP set {1.2.3.4, 4.5.6.8, 3.5.6.1} and domain B has IP set {1.2.3.4, 3.5.6.2}

S(A,B) = 0.79*(1.5/3 + 1.5/2)/2= 0.49

5.2.2 Subject Similarity between Two Domains

We also retrieve the email subject from emails that reference a certain domain. Each domain is linked to a set of subjects. The subject similarity between two domains is calculated just as IP similarity is calculated using the Kulczynski coefficient. The subject similarity score is used to strengthen the relationship between domains that partially match IP addresses. The IP similarity score can also strengthen the relationship found in email subjects.

We observed that some spam subjects are generated using patterns, for example in "Coupon ID ####", the only difference is the ID number. No common subjects will be found between these two sets of subjects using the exact match method, but we know they are related. Taking this into account, a fuzzy subject matching algorithm described in chapter 4 is used. By using fuzzy matching, the comparison between two subjects yields a score between 0 and 1, instead of a "yes" or "no" answer.

For subject set A and B, $|A| <= |B|$, we match each subject in A to all subjects in B and choose the maximum matching score S_i. The sum of S_i replaces the $|A \cap B|$ in the Kulczynski coefficient formula.

The similarity score is then calculated using the Kulczynski coefficient.

5.2.3 Overall Similarity between Two Domains

By taking the average of the hosting IP and subject similarity scores, an overall similarity score is calculated.

Forensic investigators assign the weight and threshold based on empirical experiences. When two domain names have perfect IP or subject similarity scores, we are confident these two domain names are related. Therefore, we set the threshold to be 0.5, which will cover the scenarios when the IP score is perfect regardless of what the subject score is or when the subject score is perfect regardless of what the IP score is. When the IP and subject scores are not perfect, the average score is a linear function: $x + y \geq 1$, and all the points above the line $x + y = 1$ will be accepted. We also tried the quadratic function: $x^2 + y^2 \geq 1$ and found the result was almost the same for leading

clusters, because the domain names usually have both hosting IP addresses and subjects in common.

5.2.4 Bi-connected Component Algorithm

A graph can be built by using the domain name as the vertex and the similarity score as the edge. Each connected component is initially considered a cluster. Then the bi-connected component algorithm is used to determine if the domain names in a cluster are well-connected. According to the definition of bi-connected components (Baase, 1988), a connected graph is bi-connected if and only if it contains no articulation point, also called a cut vertex. The removal of an articulation point will cause the graph to be disconnected.

The purpose of applying the bi-connected component algorithm is to determine if any domain name in a cluster acts as an articulation point. Such domain names may be popular domain names referenced by different spam emails. Therefore, a graph is constructed by connecting two domain names if their similarity score passes the threshold of 0.5, and then the bi-connected component algorithm is applied to detect any articulation points. The pseudo code of the bi-connected component algorithm can be found in Appendix C. Bi-connected components joined by an articulation point will be separated into individual clusters. But in an extreme case, when an articulation point bridges a single domain vertex from the rest graph, the articulation point is ignored because of the trivial impact on the clustering result.

5.2.5 Labeling Emails Based on Domain Clusters

Once the domains are grouped, we label the emails accordingly. However, a conflict arises if multiple domains pointing to different hosting IP addresses are found in the same email. This usually happens if a spam email references common websites, for example, "yahoo.com" or "pctools.com", etc. To deal with this, a heuristic rule is applied when labeling email messages based on domain clusters. Because a spam host contains many spam domains, a spam domain is probably connected with other spam domains. Yet a referenced URL is unlikely to be grouped with other domain names, for example, "yahoo.com" and "pctools.com" will probably stand by themselves. Knowing this, if a conflict occurs, we assign an email to the domain cluster containing the largest number of domain names. Therefore, an email is more likely assigned to the spam group rather than the referenced domain name group. The rule might not work for newsletters, but we are not interested in investigating those emails, which usually form small clusters in our experiment.

5.3 Day to Day Clustering Method

Because most leading scam groups will last for a long time, it will be worthwhile to observe the evolution of a scam through a period of time. Pharmaceutical spam is a primary example of this, several scams have spanned the entire period of this study.

Daily clustering provides a summary of daily scam groups. Next, clusters from two days can be compared based on cluster features – attributes of emails that suggest relationship between clusters. Similar clusters are considered to belong to the same scam group, which sells the same product or serves the same purpose, such as a bank's

phish or a malware spam. A cluster with no predecessor will be considered a new scam. Figure 10 shows the concept of tracing a cluster over a period of time. Nodes with the same color belong to the same scam group.

The method is: clusters of the current day are matched to the clusters of the previous day. We may focus on the leading clusters, which account for most of the spam emails that day. However, some spam may subside for several days and come back again. If a current day cluster cannot be matched with a previous day cluster, we will search the entire previous week at least before we declare a new cluster with no predecessors.

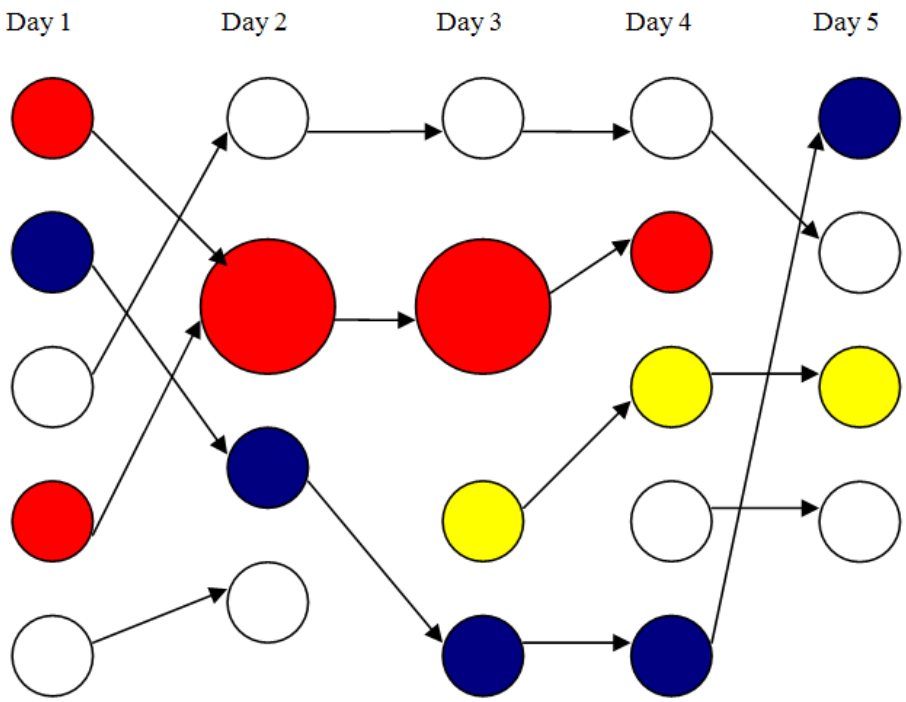

Figure 10: Multiple-day tracing of clusters

5.3.1 Similarity between Two Clusters

Two clusters are matched to each other based on the same two attributes used in daily clustering: email subject and hosting IP addresses. Each cluster includes a group of spam domain names. Each cluster is associated with a set of subjects through related emails and with a set of hosting IP addresses through domain names. The Kulczynski coefficient is used to compute the similarity among subject sets and among IP sets.

Host IP Similarity between two clusters: An intuitive way to compute the IP similarity is to find common IP addresses from the two clusters and then use a similarity coefficient. But consider the following two real clusters:

Cluster A from day 1

ip_address	domain count
60.191.221.126	327
220.248.186.101	327

Cluster B from day 2

ip_address	domain count
60.191.221.126	348
60.191.221.135	1
64.182.91.176	1
68.183.244.105	1
72.32.79.195	1
72.51.27.51	1
219.152.120.12	1
220.248.172.37	1

220.248.186.101 | 348

A daily cluster may contain many domains that are hosted at different IP addresses. Some IP addresses may host more domains than other IP addresses. In the above example, the IP addresses 60.191.221.126 and 220.248.186.101 are dominant in Cluster A and B, hosting 99% of domains in both clusters. The other IP addresses are obvious outliers, hosting only 1 domain each. It may be caused by falsified IP information or wrong inclusion of domain names in daily clustering. Simple set comparison will find poor IP overlap between the two clusters. Therefore, the domain count of each IP address needs to be taken into account.

Two IP addresses will still be matched in the same way as in section 5.2.1. Two identical IP addresses will have a perfect matching score of 1. If they reside on the same subnet (the first three octets match), a score of 0.5 is assigned.

For IP sets A and B, |A| <= |B|, we match each IP address in set A to all IP addresses in set B and choose the maximum matching score S_i. Then each matching score is multiplied by the square root of the smaller domain count of the two IP addresses. The sum of adjusted S_i will replace |A∩B| in the Kulczynski coefficient formula.

$|A \cap B| = \sum_1^n C_i S_i$ (|A| = n, |A| <=|B|), where $C_i = \sqrt{\min(a_i, b_k)}$

Here, a_i and b_k are the domain count of two matching IP addresses yielding score S_i. In the above case, the perfect matching on 60.191.221.126 and 220.248.186.101 will be counted as $\sqrt{327} \times 1$, where $\sqrt{327}$ is the adjusting coefficient and 1 the similarity score.

$|A \cap B| = \sqrt{327} + \sqrt{327}$

Instead of using the number of IP addresses as the set size, the sum of square roots of all the domain counts in the set is used as the set size. If a_i is the domain count for an IP address in cluster A and b_i is the domain count for an IP address in cluster B, then

$$|A| = \sum_{i=1}^{m} \sqrt{a_i} = \sqrt{327} + \sqrt{327}$$

$$|B| = \sum_{i=1}^{n} \sqrt{b_i} = \sqrt{348} \times 2 + 7$$

S(A,B) = Kulczynski(A, B) = (|A∩B|/|A| + |A∩B|/|B|)/2

= (1+36.17/44.31)/2=0.9

Subject Similarity between two clusters: The subject similarity between two clusters is computed in the same way as the subject similarity between two domain names in daily clustering (See Chapter 4 and section 5.2.2). The cluster with the fewer subjects is matched to the other cluster using fuzzy string matching. For each subject in the smaller cluster, the best match is found in the larger cluster. Then the summation is taken as the intersection. The Kulczynski coefficient is used to capture the subject similarity of the two clusters.

5.3.2 Linking Two Clusters

The average of subject and IP similarity scores between two clusters is used to decide whether the two clusters are related. Because the two clusters are from different days, we relax the threshold a bit. However, we are unsure how much we shall relax because it is difficult to predict when the spammer will make major changes to his spamming strategies. Therefore, as long as they are not zero, we will store all the

similarity scores in the database. The investigator has the choice to set a threshold to select the scores that interest him. For the experiment, we set the threshold to be 0.4 for clusters from adjacent days, a littler lower than the threshold for daily clustering.

5.4 Experimental Results

For this study, 638,678 email messages were collected in the months of June and July of 2009. The emails were contributed by the same volunteer who provided data for the previous two experiments. From the 638,678 emails collected, 350,394 emails were used for clustering. The remaining emails were excluded because the parsing program did not find a URL in the emails or the domain name extracted from the URLs could not resolve to an IP address, indicating the advertised website was unavailable.

We extracted 16,348 domains from the emails, and most used wildcard DNS entries. The ratio between the number of hostnames and the number of domain names is over 100: 1. The hostname here is a sub-domain created from an existing domain by attaching a string before the domain name. For example, "live.com" is a domain name, and "ghl234.live.com" is a hostname. This indicates that by studying domains instead of URLs in emails, we can effectively compress the data while not losing valuable information.

5.4.1 Daily Clustering Results

Most daily clusters are very small, containing at most six emails and at most two domain names. The largest daily cluster usually has more than 1000 emails and more than 100 domain names. Figure 11 shows the number of emails in the top 5 daily

clusters compared to the total number of emails used in clustering. The emails in the top 5 daily clusters account for about 83% of total emails.

The leading clusters are most interesting to us and probably also to law enforcement personnel. Therefore, we further examine the large clusters to determine if the domains and emails in those clusters are really related. For example, the largest cluster on July 30 has 2617 emails and 155 domains, which account for almost 48% of the emails included in clustering that day. This shows how dominant the leading clusters are in our dataset.

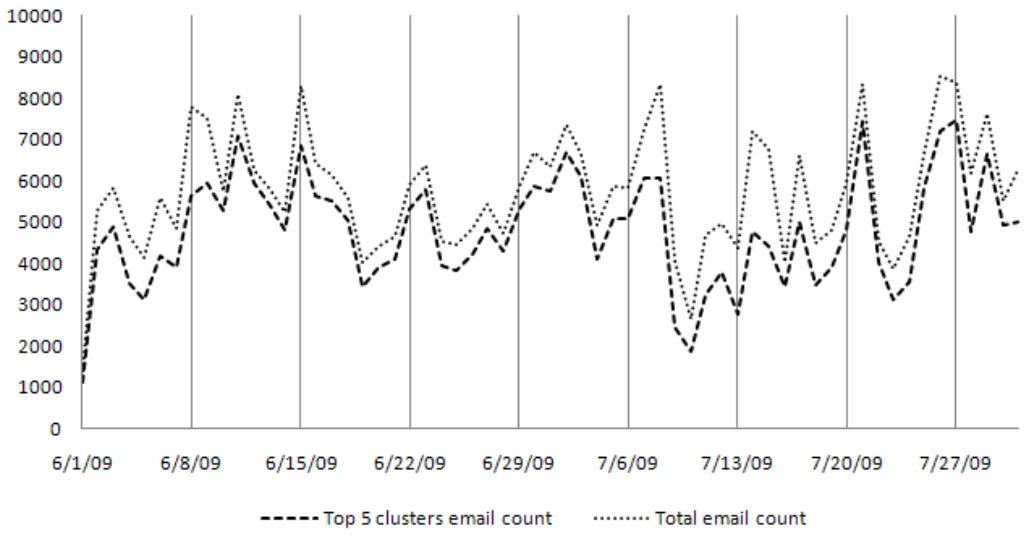

Figure 11: The number of emails in top 5 clusters compared to total email Count

Figure 12 shows the interconnectivity of domain names, hosting IP addresses and the country of the network containing the IP addresses in the largest cluster on July 30. The domain names are connected to the IP addresses and IP addresses to the countries. The 155 domains are divided into three subgroups based on hosting IP addresses. The biggest sub-group contains 140 domains, which were all hosted at four dominant IP

addresses. The second sub-group contains 13 domains, which were hosted at several other IP addresses in addition to the four main IP addresses. The third sub-group contains only two domains: one is hosted at five IP addresses, out of which four are common to the ones in sub-group 1, the other is hosted at 159.226.7.162. Three of the four dominant IP addresses reside in China and the other in Russia. It is unlikely for an investigator to relate an IP in Russia to IP addresses in China unless there is sufficient evidence to support that.

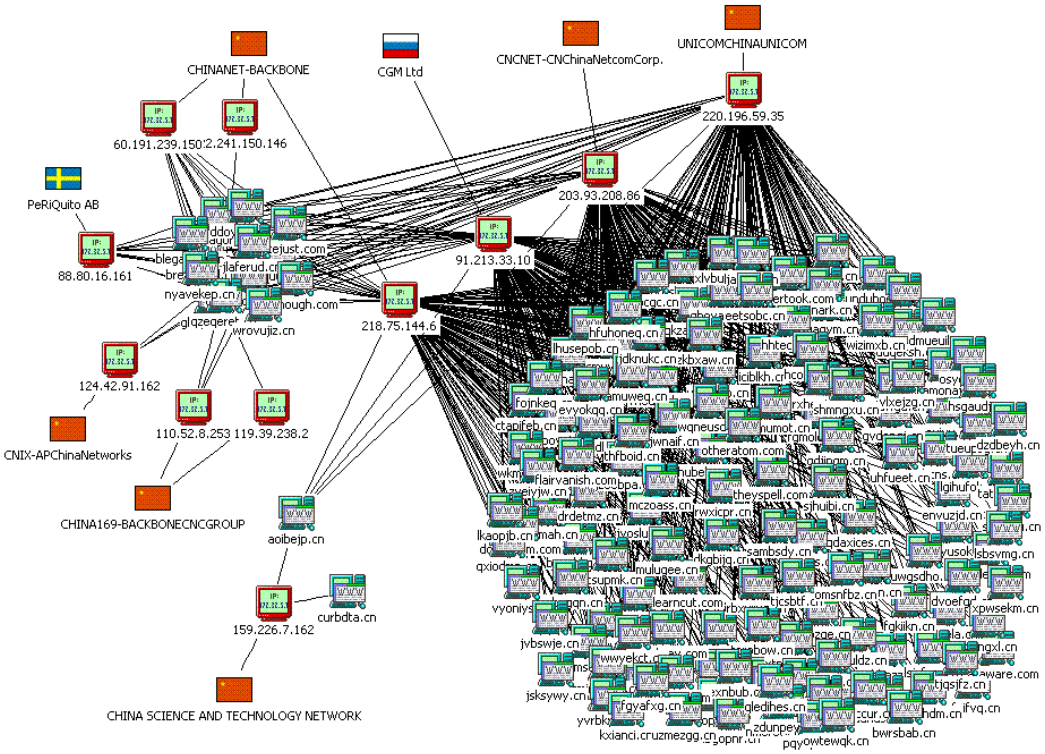

Figure 12: Domains and related IPs from the largest cluster of July 30, 2009

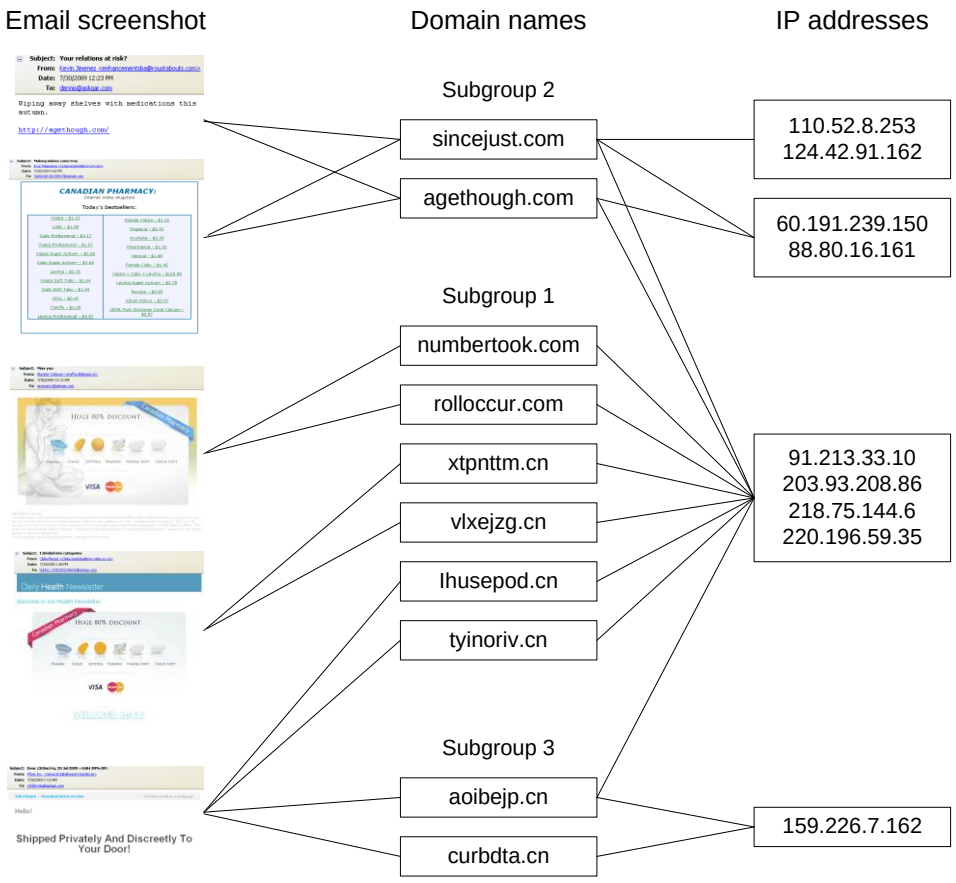

Figure 13: Relationships among sample emails, domains and hosting IPs from the largest cluster of July 30, 2009

We then pulled email samples for several domains in each of the sub-groups. Figure 13 showed the connection between some sample emails, domain names and hosting IP addresses. Sample domain names were taken from each subgroup from Figure 12 and put into the middle column. The first two domains were sampled from the second sub-group, the last two from the third sub-group and the rest from the largest sub-group. Sample email screenshots were taken for the 10 domain names and put into the left column. The associated hosting IP addresses were put into the right column. The links showed that they were all related to each other: they either shared the same host IP addresses or were referred to in emails with the same template. Subgroup 2 was linked to

subgroup 1 by the common hosting IP addresses. Subgroup 3 was linked to subgroup 1 by common emails. We could see at least four different email templates that substantially differed from each other in appearance. A human may still be able to link sample email #3 with #4, but is unlikely to link #1, #2, and #5 together. Several more email templates from the largest cluster were not illustrated here.

We checked sample domains in the three subgroups and they were all “Canadian Pharmacy” scam websites. We believe the remaining domain names are likely also “Canadian Pharmacy” scam. The fetched web pages may also group these domain names together, but the process is time-consuming and the fetched web content may be incorrect. For example, some hosts have counter-measures which ban an IP if an agent from that IP tries to probe the server repeatedly. If we continued to probe, we would eventually get a time-out response.

The business model of affiliate program spammers raises further concerns. Many large affiliate programs, for example the GlavMed program, which owns the illegal "Canadian Pharmacy" content, pay individuals for creating traffic which results in purchases of their products. On one level, all of the “Canadian Pharmacy” spammers are related, because they are all spamming members of the GlavMed affiliate network. However, it is more valuable to identify the spammers by their individual organizations. A familiar example of affiliation is the franchise program for a large fast-food restaurant such as McDonalds. Some McDonald's franchisees own just one restaurant while others own several dozen. Not all McDonald's restaurants are owned by the same company, but they are affiliated. In addition, a restaurant franchisee may own many kinds of restaurants, not just McDonald's. In the same way, a spammer may spam

for several different programs, one may send spam for pills and watches, while another sends spam for pills and pornography. By concentrating on what spam is sent, and where the spammed websites are hosted, we believe we are identifying the "franchisee", rather than making the error of grouping together all spammers who belong to the same affiliate program. In clusters from other days, we see websites such as counterfeit Rolex watches, Canadian pharmacy and Bank of America phishing mingled together.

5.4.2 Tracing Clusters over the Experiment Period of Time

In this experiment, we traced clusters from adjacent days for two months. A threshold of 0.4 is used: if the average IP similarity score and subject similarity score passes that threshold, the two clusters are considered related. The biggest cluster is traced from the beginning of June to the end of July, with average IP score of 0.89 and average subject score of 0.28.

Figure 14 shows the number of emails and new domain names belonging to the biggest cluster for the experiment period. Here new domain names means the domain names have never been seen in our database prior to the current date. Therefore a domain name will only be counted at the date when it first appeared no matter how long it lasts. The total number of emails is 221,654, compared to 7,386 domain names. There were no new domain names found on July 16, even though the spam emails kept coming; maybe the spammers took a day off.

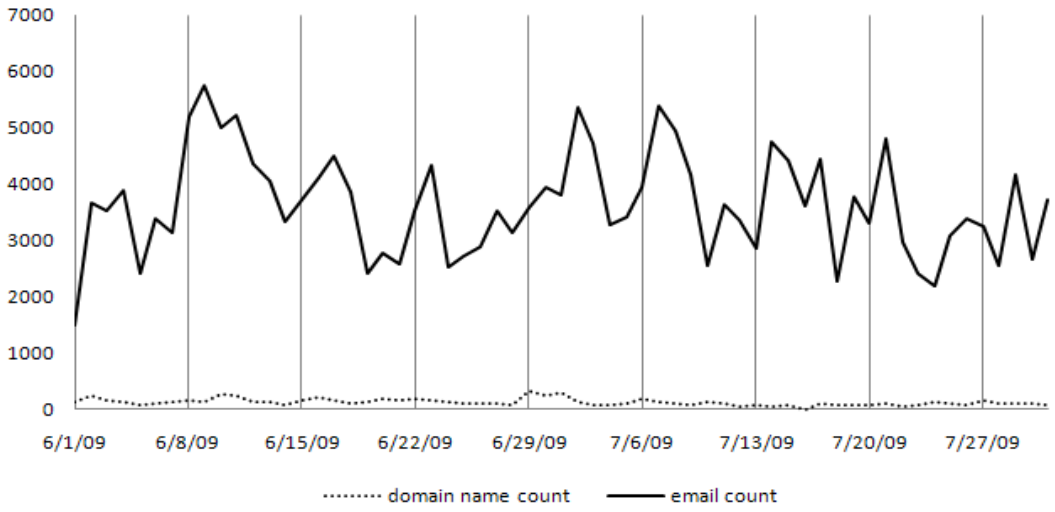

Figure 14: Daily email and new domain count of the largest cluster

Table 3: Domain count of top-level domains in the largest cluster

Top-level domain	cn	com	ru	com.cn	net
Domain count	9107	1029	303	26	2
Domain lasting for one day	5538	400	229	2	1
Percentage of domains lasting for one day	61%	39%	75%	8%	50%
Period seen	6/1 – 7/31	6/1 – 7/31	6/1 – 6/11	6/4 – 7/15	7/22 – 7/29

Most domain names last for a short time, 59% lasted for one day and 39% lasted for two days. If we break down the domain names by their top-level domain, most of the domain names end with ".cn", followed by ".com", with the ".com" count being only 10% of the ".cn" count. In Table 3, the second row shows the number of domain names for each top-level domain in our largest cluster; the third row shows the number of domain names which last for only one day for each top-level domain; the fourth row is the

percentage of row 3 in proportion to row 2; the last row is the time period in which each top-level domain appeared. The ".com" domain names usually live longer than ".cn".

Table 4: Top hosting IP addresses of the largest cluster

IP address	Host owner	Country	Active period	Domain count
58.17.3.41	China Beijing Superman Internet Cafe	China	5/27 – 6/21	3427
60.191.221.123	Jinhua Telecom Co., Ltd	China	6/10 – 6/21	1965
60.191.239.150	Jinhua Telecom Co., Ltd	China	7/1 – 7/26	1947
60.191.239.153	Jinhua Telecom Co.,Ltd	China	6/20 – 6/28	1008
61.191.191.241	Hefei Chinanet Anhui Province Network	China	6/10 – 6/30	2779
119.39.238.2	Cnc Group Hunan Yueyang Network	China	6/20 – 7/5	1965
203.93.208.86	Qingdao China Unicom IP Network	China	5/22 – 7/31	7600
218.75.144.6	Changsha Chinanet-hn Changde Node Network	China	6/20 – 7/31	3861
222.241.150.146	Changsha Chinanet-hn Hengyang Node Network	China	6/29 – 7/5	1051

The biggest cluster contains 42 IP addresses, 14 of which have more than 1000 associated domain names. Some IP addresses appeared in our database as early as late May and some were still active in August. Table 4 shows some of the top IP addresses (associated with most domain names). They are located on different networks in China. Some IP addresses are used for a short time, but IP 203.93.208.86 is used throughout the

experiment period. IP 58.17.3.41 is used in the first half, stopping at June 21 and IP 218.75.144.6 picks up in the second half, from June 20 to July 31.

An interesting observation is the correlation between different IP addresses on the number of associated domain names. This occurs because when a new spam domain appears, it usually points to several IP addresses. As a result, we see a high correlation on the domain name count among IP addresses over a period of time. In this case, only domain names never seen before will be counted.

Figure 15 shows the correlation between 58.17.3.41 and 203.93.208.86 from June 1 to June 19, and the correlation between 218.75.144.6 and 203.93.208.86 from June 22 to the end of July. June 19 to June 22 appears to be the transition period when the DNS entries are being updated.

Correlations between short-lived IP addresses existed as well. Figure 16 shows the correlation between 218.75.144.6 and two other IP addresses during the second half of the experiment. IP 218.75.144.6 is perfectly correlated with 119.39.238.2 from June 20 to July 5, and perfectly correlated with 60.191.239.150 from July 8 to July 22. Even though the spammers are moving domains among different IP addresses, some IP addresses are more consistent. Before an IP is totally discarded, some domains still point to that IP address. Therefore we were able to find partial IP overlap between clusters of adjacent days during the time of IP shifting.

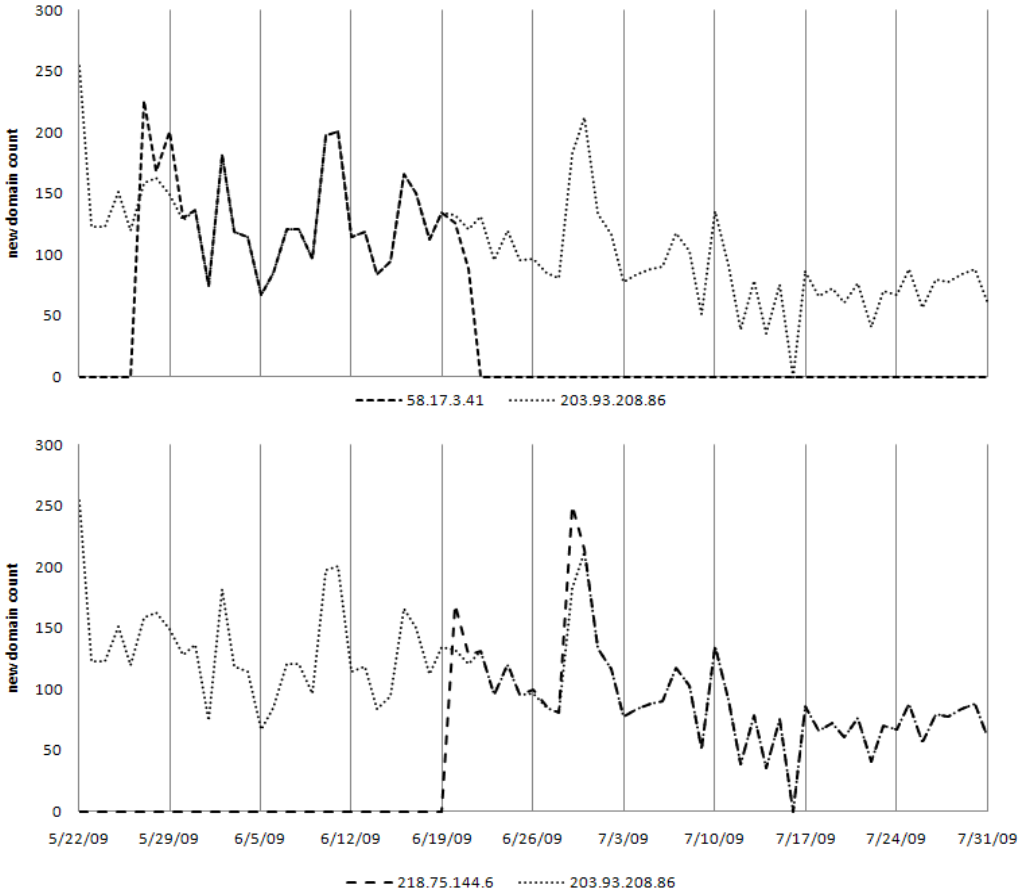

Figure 15: The number of new domains hosted on IP addresses 58.17.3.41, 218.75.144.6 and 203.93.208.86

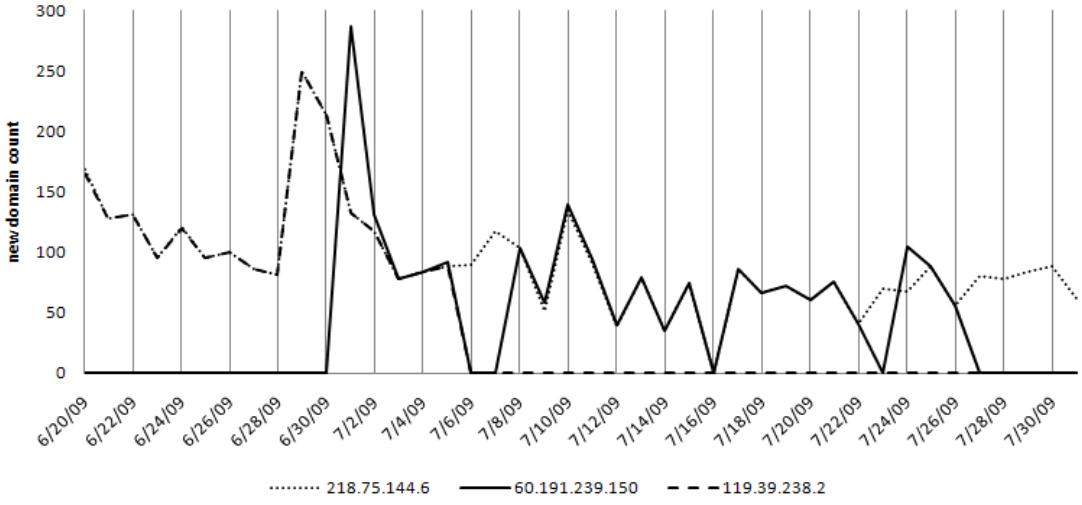

Figure 16: The number of new domains hosted on IP addresses 218.75.144.6, 60.191.239.150 and 119.39.238.2

We also checked IP addresses of the sending machine, located in the "Received" records of email headers. In the largest cluster, the number of sending IP addresses is about 70% of the number of emails. The number of sending IP addresses increased and decreased along with the number of emails (Figure 17). The sending IP addresses are evenly distributed among different IP ranges, thus the spam emails are coming from all over the world.

When the number of spam emails increased on some days, it was because more machines were sending spam, not because some machines were sending more emails. The large number of sending IP addresses suggest that the spam in the largest cluster is probably sent via botnets. Therefore, the spammer who created the web sites is likely responsible for spreading Trojan viruses and turning computers into bots, or does business with the botnet creator.

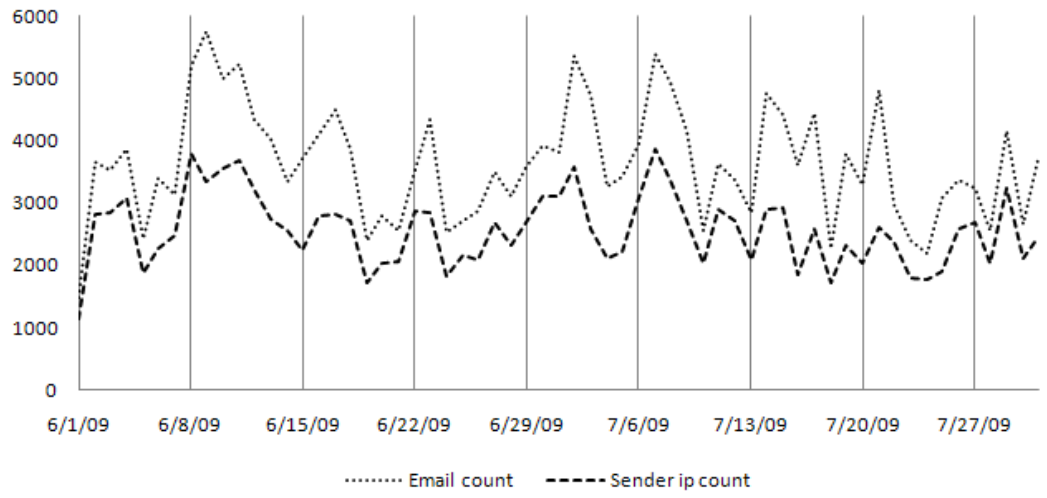

Figure 17: The number of emails and sending IP addresses in the largest cluster

5.5 Discussion

Starting with spam, we investigated the domain names appearing in emails and their hosting IP addresses, combined with email subjects. We are able to link spam messages that are seemingly unrelated based on human observation of their inherent attributes (Figure 13). The biggest clusters account for one-third to half of daily spam emails. Based on human observation, the spam is mostly pharmaceutical spam.

The results of this experiment verify that, to combat domain blacklisting, spammers register a large number of domain names. The largest spam group we found was associated with almost a hundred new domain names each day. The registrar records of some domain names in our cluster showed a single identity registered hundreds of domain names in a short time and the identity was obviously a disguise. For example, a Chinese interior remodeling company registered over 100 domains, and many of them were Canadian Pharmacy scam domains. It is unlikely that an individual company will register so many domain names for legitimate purposes and the content hosted there is unrelated to the company's business. Many domains were registered in China, ending with ".cn". By checking the destination web sites, we found many ".cn" domains were redirected to ".com" domains when the website was visited. Therefore, the short-lived domains serve to protect the real destination domains which never appear in our collected spam emails. We suspect the ".cn" domains are easy and cheaper to register and the registrar does not care how the domain names are used.

The 7,000 plus domain names found in the largest cluster in the experiment were linked to 221,654 emails. Even though many emails had different appearances in email bodies (Figure 13), we believe they are related to one spam group because the destination

web sites have the same look and feel. Just using inherent attributes from emails, such as email content and email header, would fail to group them together.

The spammers also exploit wild-card DNS records to create numerous phantom host names from a single domain name. This suggests domain name blacklisting will be more effective than URL blacklisting if we can confirm that a domain is registered solely for spam usage. It also explains why Webb et al. (2007) found many duplicate web spam pages in their corpus. If so many host names are created from a relatively smaller set of domain names that are actually hosted at the same place, it is not surprising the fetched web pages will be identical. Some of the domains in our cluster were associated with more than 10,000 hostnames. Therefore, considering today's volume of spam, fetching the web pages for all of them would not be efficient.

By monitoring the hosting IP addresses, we discovered several networks that are heavily used by spammers, mostly residing in China. The lack of adequate regulation and legislation (Qi, Wang & Xu, 2009) in that country encourages spammers to exploit the networks there. Fourteen IP addresses have been found to host more than 1000 domain names. As a way of load balancing, spammers register numerous domain names and point each to several IP addresses. Domain names created during the same time will show a high correlation of hosting IP addresses. From time to time, the spammers will redistribute domain names to a new set of IP addresses. However, some IP addresses remain active longer, allowing us to link new IP addresses to old IP addresses. The results suggest that new spam domains can be effectively detected by checking their hosting IP addresses and significant hosting IP addresses can be reported to law enforcement personnel for termination.

The day-to-day tracing of clusters has some limitations. Only the largest cluster is traced successfully through the two-month period of time. The traces of other clusters were lost after a few days. However, after a day or two, new clusters emerge, and according to human observation, they resemble old clusters. The biggest cluster is well-traced because spam emails keep coming every day. For other clusters, the spam emails may stop for a day or two.

Moreover, spam investigators may prefer to cluster spam emails using a shorter time interval because they want to promptly detect new spam domains. Then an hourly, or even a quarter of an hour interval, may be preferred over daily clustering. Tracing clusters at shorter intervals will be harder than tracing daily clusters because certain spam may only come at specific hours or the time interval can be irregular. Therefore, a more rigorous tracing algorithm is needed so that a cluster can be matched to old clusters at random intervals. Each cluster should be associated with a time stamp, which can determine if the cluster is recent enough to be considered in clustering.

6. TRACKING CLUSERS USING HISTORICAL DATA

In the previous chapter, we discussed an algorithm for clustering daily spam domains and linking related clusters across adjacent days. In the real world, it is desirable to cluster spam domains of a shorter time interval so that immediate action against the spam threat can be taken. The daily clustering algorithm can be easily modified to cope with shorter time intervals by changing the window size of the input data. But we need to find a new approach to trace clusters over a period of time. The comparison between adjacent daily clusters may work for major scam groups, because they are expected to come every day. But once we begin to cluster on a shorter time interval, for example an hourly basis, the comparison of adjacent hourly clusters may find nothing because a particular spam may not appear every hour. Even if the algorithm can be modified to trace further backward, there is a problem of inefficiency, because we do not know how far we must trace back before a predecessor can be found if it is even found. A lot of time may be wasted searching for a possible match when there is no match.

In this chapter, we propose a new framework for tracing clusters over a long period of time. The concept is similar to the data stream clustering in which a newly-formed cluster is absorbed into a historical cluster most similar to it, given the similarity score between them surpassing a threshold. If no match is found, it will be declared as an emerging cluster. When a new cluster is merged with a historical cluster, the historical

cluster will be updated to reflect the new information. A time threshold will determine how far we want to trace back in the historical repository.

6.1 Historical Cluster Repository

A historical cluster repository will be built to store interesting clusters resulting from single-time-slot clustering, which uses the same algorithm as the daily clustering. The change can be made to the input module, which will retrieve data on the required time interval. The cluster repository will consist of several tables that store relevant information on historical clusters.

The main table will contain the cluster ID, the time when it is first seen and the time it is last seen. Additional tables will be created for the attributes that are useful for tracing clusters. The IP table will contain the cluster ID, the hosting IP addresses of spam domains, the time when the IP address was first used and the time it was last used. The subject table will contain cluster ID, distinctive subject patterns, the time first seen and the time last seen. Instead of storing each subject, we will store the subject pattern because a cluster may contain a variety of customized subjects. Storing individual subject wastes storage space and slows the comparison of clusters. The subject pattern can be extracted manually or using the algorithm described in section 4.3.

To find the best match, when a cluster is formed, its hosting IP addresses and subject patterns will be compared with historical clusters. The IP and subject similarity score will be used to measure the similarity between two clusters. The similarity threshold is subject to change based on experimental results. It is expected to be lower than the threshold used for daily clustering. Spammers will introduce new IP addresses

and subject patterns from time to time. Therefore, the link during the changing period of time will be weaker than when there is no change. During the transition time, we may still see some overlap when the old IP addresses and subject patterns are used along with the new IP addresses and subject patterns.

A historical cluster can be considered obsolete if the time last seen is old enough. Forensic experts can determine if that period should be two weeks or a month. An IP address and a subject pattern can be marked as obsolete and excluded from clustering if it has not been seen for some time while the cluster it belongs to is still active.

6.2 Experiment on IP Tracing

We have not implemented the tracing of clusters using historical data. That will be part of our future studies. A preliminary experiment was conducted using the IP table to trace clusters over one week. Two clusters from different timestamps were linked if they shared a common hosting IP address. Some IP addresses were found to last for more than a month.

For this experiment, spam emails were gathered from a volunteer ISP with more than 10,000 email accounts. The emails were sent to addresses that were no longer active, implying most of them were spam. The dataset contains 35 million email messages in January 2010. About 95% of the emails contain URLs. The domain names were extracted from the URLs.

The spam domains from January 1 to 8, 2010 were used for clustering on an hourly basis. There were on average 500,000 emails and 1000+ domain names per hour. Each hour there were about a thousand clusters formed, most of them being small clusters

with less than 100 emails. We suspect those are newsletters and promotional emails sent by advertising companies. The clusters containing over 100 emails were used to further investigate hosting IP addresses. There were about 30 – 60 clusters in that category per hour. The email and domain count were then used to identify the most dominant spam clusters and corresponding hosting IP addresses.

6.2.1 Canadian Pharmacy Scam

The most significant cluster is the Canadian Pharmacy (CP) scam, which is identified by fetched websites. It contains domain names that are primarily hosted at 60.172.229.102 and 61.235.117.75, with a few hosted at 58.218.199.97. The scam became dominant on the 7th hour of Jan 3, and accounted for about 14 thousand emails during that hour. Spam emails of this scam were seen every hour after that.

Figure 18 shows the hourly email count starting from the 7th hour of Jan 3 until the end of Jan 8. The dotted line showed there were about 50,000 – 60,000 emails per hour. The solid line illustrated the change of email volume of CP scam. At some hours, the CP scam accounted for over half of the total amount of emails. Even when the volume is low, the CP scam still accounted for about 10% of the total emails. Surprisingly, while the CP scam volume went up and down, the total spam volume remained stable, indicating the volume of other scam groups increased when CP spam volume declined.

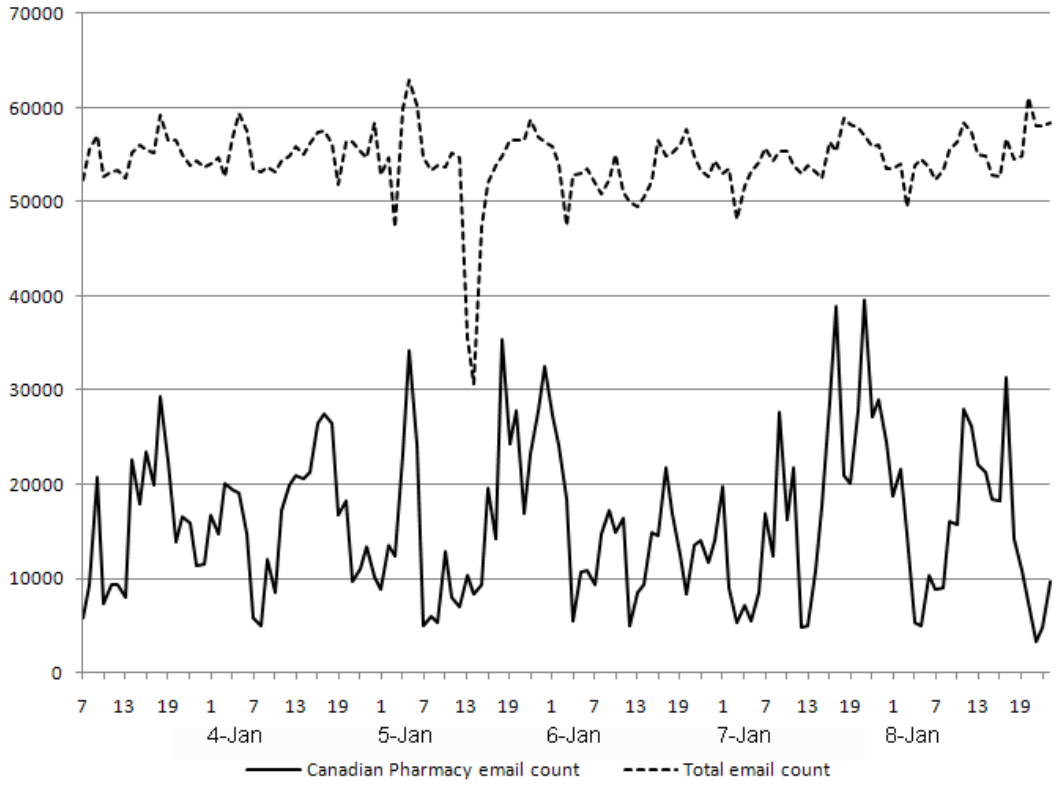

Figure 18: Hourly email count of Canadian Pharmacy scam comparing to total email count, Jan 3-8, 2010

In most days, our clustering algorithm separated the cluster into three sub-clusters because the spammers use three different subject patterns (the words in bracket are customized):

Notification to [username] special 80% OFF of Pfizer

Special 80% discount for customer [name] on all Pfizer

Valued customer [email address] 80% OFF on Pfizer.

Since there is only one IP address, the adjusted coefficient (described in section 5.2.1) will generate an IP similarity score of 0.5. Different subject patterns will force the domains into different clusters. Human investigation of the subject patterns and fetched

web pages confirm they are one scam. The emails were probably sent by different botnets using different templates.

Before Jan 6, all domains in that cluster were hosted at 60.172.229.102. On Jan 6, the domains were hosted at both 60.172.229.102 and 61.235.117.75. On Jan 7, more domains were hosted on 61.235.117.75 than 60.172.229.102. On Jan 8, no domains names were found to be hosted at 60.172.229.102 in our database. The IP 61.235.117.75 was traced all the way to the first week of March in our database before being replaced by a new IP address (Figure 19). Both IP addresses are located in China.

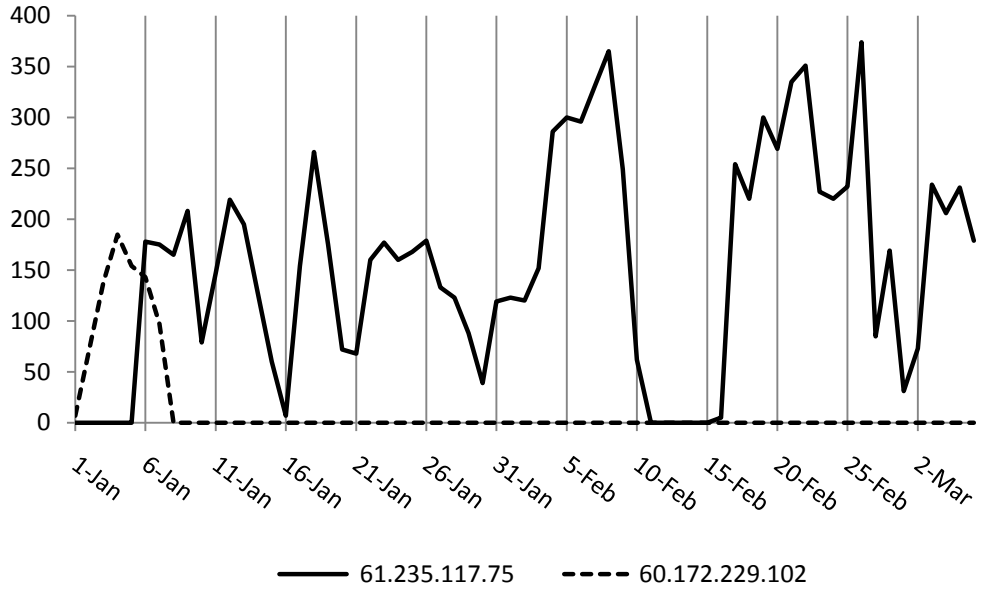

Figure 19: The number of new domains hosted at IP addresses 61.235.117.75 and 60.172.229.102, Jan 1 - Mar 6, 2010

Starting from Jan 7, there were 9,921domain names hosted at 61.235.117.75 that were related to Canadian Pharmacy scam. The domains have either ".cn" or ".ru" as the Top-level-domain (TLD), indicating they were either registered in China or Russia. The two dominant patterns in the domain names were: (1) concatenation of two English

words, such as “senseleast.ru”; (2) alternation between constants and vowels, such as “quzixenov.cn”. The “.cn” or “.ru” domains were redirected to a “.com” domain. For example, “cottonwe.ru” was redirected to “pillsgreatenter.com”. Both domains are hosted at the same IP addresses, with most current DSN records pointing to 202.111.175.31 and 218.93.201.53. The WHOIS information shows that a person named “Zhaohua”, a very common Chinese name, registered “pillsgreatenter.com” on Jan 8. There have been only 2 changes on NS records and 3 changes on IP A records since then. The same person owns 725 other domains. The reverse IP records show that 2063 other domains are also hosted at the same server. This means that the IP record change is infrequent and the same IP address can be used to detect many other spam domains hosted there.

Among the 9,921 domains we discovered, 5,722 domains showed up in spam emails for only a single day. This means that even if human investigators reported them to domain blacklists, one would probably never see these domains again. However, if we use the hosting IP address to blacklist the domains, those 9,000+ domains can be easily detected as spam domains.

6.2.2 Ultimate Replica Watches Scam

Another interesting scam is the Ultimate Replica Watches. Those domains were hosted at 116.127.27.188, located in South Korea. However, other domains hosted at the same IP address pointed to sexual enhancement websites. In some hours of a particular day, the clustering results show 2 to 3 clusters all hosted at that IP address. The clusters were differentiated by the email subjects. For example, on the 9th hour of Jan 3, one

cluster was about replica watches, another about penis enlargement, the third was DrMaxman (also a sexual enhancement product).

The IP address 116.127.27.188 was traced till Feb 16 (Figure 20). This cluster is much smaller than the Canadian Pharmacy scam, averaging about several hundred spam emails per hour and 20 new domain names per day. However, on Jan 18, a burst of domain names appeared on that IP address.

Spam emails from this cluster did not continue for the whole day. They were seen for a few hours, then disappeared, and then returned a few hours later.

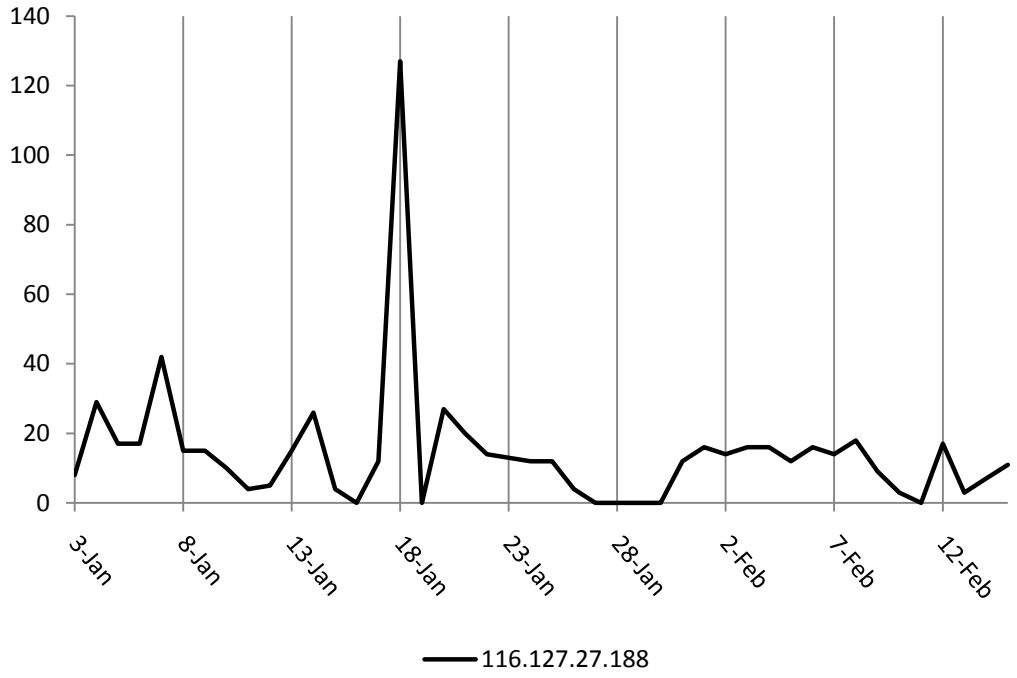

Figure 20: The number of new domains hosted at IP address 116.127.27.188, Jan 3 - Feb 16, 2010

6.2.3 Tracing a Phishing Campaign

Phishing spam is quite different from pharmaceutical spam, which lasts for a very long period of time. Phishing spam comes and goes. The web links used by phishing spam are usually short-lived. The domains used by phishing can resolve to a large number of IP addresses, probably supported by FFSN. Each IP address has a short TTL value. DNS lookups of a phishing domain at different time will resolve to different IP addresses.

Our clustering results initially identified six outstanding IP addresses serving the same spam domains. The DNS record lookup returned all six IP addresses for several domains in that cluster. Tracing the clusters containing any of these six IP addresses identified the emergence of a phishing campaign.

Table 5 shows the timestamp and the corresponding number of resolved IP addresses found in related clusters. Starting from Jan 6, 9am, there were only 21 IP addresses identified. At Jan 7, 8am, the number jumped to 198. At Jan 8, 8am, it burst to 455, then at 12pm, to 476. The email count was also on the rise, but not as dramatically as the IP count. The number of IP addresses began to subside after Jan 9.

Table 5: The number of IP addresses used by the phishing campaign

Time stamp	Jan 6, 9am	Jan 7, 8am	Jan 8, 8am	Jan 8, 12pm
IP count	21	198	455	476
Email count	233	264	618	1032

The following is a sample email message:

Dear user of the email.com mailing service!
We are informing you that because of the security upgrade of the mailing service your mailbox (username@email.com) settings were changed.
In order to apply the new set of settings click on the following link:
http://email.com/owa/service_directory/settings.php?email=username@email.com&from=email.com&fromname=username
Best regards, email.com Technical Support.
Letter_ID#B2S602QQE9P3X

The actual URL points to a domain "okqwac.com.pl" instead of "email.com". With the URL already disabled, we are not sure if it was used to steal personal information or to spread a virus. (Note: every user received a customized email, we substituted the real username with "username" and real domain name with "email.com").

6.2.4 Other Scams and IP addresses

Apart from the IP addresses described in the previous sections, we also discovered several additional IP addresses worth noting. Table 6 shows some significant IP addresses identified during the first week of January. These IP addresses did not have the longevity of the Canadian Pharmacy IP addresses, but during their active time, they accounted for a significant number of domain names and emails. During some hours, the total related spam emails surpassed 10,000.

Table 6: Summary of other significant hosting IP addresses

IP addresses	Domain count	Active period	Products
124.61.222.223	711	Jan 3 – 8	Watches and sexual enhancement
58.218.250.107	255	Jan 1 –28	Drugs
202.111.175.126	68	Jan 3 –5	Sexual enhancement
116.123.221.91	61	Jan 3 – 7	Drugs Casino

6.3 Discussion

This experiment used a larger dataset than the one used in chapter 5. The spam emails were collected from many email addresses rather than a single "catch all" domain and the spam emails reflected more diversity than the previous dataset. Spam domains extracted from emails were clustered on an hourly basis and clusters were traced using similar IP addresses.

The results show that IP tracing will be effective against pharmaceutical, sexual enhancement and luxury good spam, which mainly use static IP addresses to host the websites. Domain names clearly outnumbers hosting IP addresses. The IP addresses remain active from several days to even a couple of months. The spammers keep registering new domains to replace old domains. But because the hosting IP addresses did not change, new domains can be easily linked to old domains. Occasionally, spammers change the hosting IP addresses. But there was a short period of time when both old IP and new IP addresses were used, allowing us to link the new IP address to the old IP address.

The results can be used to improve the effectiveness of domain blacklisting. Current domain blacklists are maintained manually by collecting domain names from

spam emails. By the time they are listed, many domain names are inactive. The sending IP blacklist is not effective against botnet-generated spam because the blocked IP addresses are actually victimized computers. The IP of the botmasters are usually well-protected. The hosting IP blacklist can be used to detect and shut down new spam domains. IP addresses hosting many spam domains can be identified and can check domains in newly-arrived spam emails. The domain names and IP addresses can be further investigated to find the content of the website, who registered the domain names, the location of the IP address and what other websites are hosted there. Law enforcement personnel can use this information as evidence against ISPs which provide aid to spammers.

The method may be ineffective against phishing spam, where IP addresses outnumber domain names. Hacked domains, blog spaces and infected computers redirect users to the real phishing site, because investigators vigorously pursue phishing sites and try to shut them down. However, counting IP addresses can reveal an emerging phishing spam, even though terminating it may require additional effort.

On the other hand, the sites that sell drugs and sexual enhancement products are seldom touched, making it unnecessary for spammers to change the hosting IP addresses frequently. They only need to flush out old domain names blacklisted by spam filters. Therefore, blocking hosting IP addresses will still be effective against point-of-sale spam websites and associated spam emails. The pressure from the investigators may push the spammers to use more advanced techniques, such as the FFSN, to protect their hosts. Nevertheless, this will raise the operation cost and reduce the effectiveness of spam.

7. CONCLUSION AND FUTURE WORK

This research aims to apply data mining techniques to assist the termination of spam emails. While spam filtering has been the major weapon against spam for many years, evolving spamming tactics and strategies have rendered blocking spam at the user's end no longer effective. Spam filtering does not reduce the number of spam messages. To win the war against spam, we need to cut the spam "supply line," where spam is produced and profit is generated. Fighting spam at the source, where the botmaster and the hosting servers lie, can deliver a more powerful blow to the spamming business than filtering.

To accomplish this, we must understand how a spamming network operates. A botmaster, known as a C&C server, controls the operation of bots, which send spam. The hosting servers are used for hosting spam websites and serving as nameservers for spam domains. The shutdown of C&Cs can disable the botnets. The termination of a hosting server can substantially cripple the spam websites and cut the generation of revenue for some time. For example, termination of the McColo, a rogue ISP which provides hosting service to spammers, caused spam volumes to drop by 36% in the United States, and by as much as 73% in other regions of the world (DiBenedetto et al. 2009). In a month, the spam volume returned to its previous level, indicating that efforts to target spam hosting servers must be ongoing.

Initially, our approach tried to cluster spam emails that share a common attribute. Later, a fuzzy string matching algorithm was introduced to allow partial matches between

emails that are generated by templates. The spam domains in the emails turn out to be a more interesting target than the email itself because they lead to the hosting servers controlled by spammers. Therefore, a derived attribute, the hosting IP address of spam domains, was included in the clustering along with the email subject. The clustering method proved to be useful against large scam groups that use many spam domains and wildcard DNS records to combat URL blacklisting. The domains are hosted at the same set of IP addresses for some time. According to the domain registrar information, the domains are usually registered by a phony company or person, who owns numerous spam domains. From time to time, the spammers will switch to a new set of IP addresses. However, there is still an overlap between the IP addresses during the transition period.

Some domains are served by FFSN, which use bots as a layer of protection for the real hosting server. The DNS lookup on a FFSN domain will return the IP address of a bot, which serves as a redirection point to the hosting server. Each bot will serve the domain for a very short period of time, usually less than a minute, making it difficult to trace the real server. More and more FFSN domains have been spotted, mainly in phishing emails (Gupta, 2008) and the use of FFSN is still not prevalent in spam hosts, because a FFSN is more difficult to maintain and spammers will only adopt this strategy if the spam domains are in danger. The phishing domains are more vigorously pursued by anti-spam groups than other spam hosts, therefore more FFSN domains are seen in phishing emails. Our research found static IP addresses were still heavily used by point-of-sale spam websites, which include pharmaceutical, luxury goods, casino and sexual-enhancement spam. A limited number of IP addresses accounted for thousands of spam

domains in less than a month and the related spam emails contributed the majority of the spam messages in our dataset.

7.1 Benefits and Impact

The research proposed a useful framework for clustering and detecting significant spam domains and their hosting IP addresses from a large number of spam emails. The results will benefit the anti-spam practice in several ways.

7.1.1 Improving Domain Black Listing

Most popular Domain Block Lists (DBLs) are generated through the collaborative efforts of spam investigators, who manually or with the aid of simple programs identify spam domains from spam messages. This is inefficient considering the massive volume of spam. Moreover, spammers can keep registering new domains to replace the ones that have been blocked. Sheng at al. (2009) found 63% of the phishing URLs lasted for less than 2 hours while on average it took 12 hours for a phishing URL to show up on the blacklist. Likewise, most of the pharmaceutical spam domains identified in our research appeared in spam emails for less than two days. Over 60% of the domains with ".cn" as TLD were used for only one day and never seen again. Wildcard DNS records allow spammers to create an infinite number of phantom hostnames from registered domains, making the URL blacklisting even less effective. Many blacklisted URL and domain names will never be seen in spam emails and they are of little use in blocking further spam messages of the same kind.

However, because the new domains still resolve to the same IP address, they can be easily identified as spam by comparing their hosting IP addresses with blacklisted IP addresses. Most current domain blacklists do not provide this function, for example, the Spamhaus DBL, maintained by a dedicated team of experts, is actually a domain URI block list and does not support IP lookup (Spamhaus DBL, 2010). Therefore, DBL can be significantly improved by using a hosting IP blacklist.

Our approach is effective against spammers who use a large number of domains, but a limited number of hosting IP addresses. The hosting IP blacklist can be used to detect and block new spam domains. Our tracing algorithm can pick up new IP address during the transition period. Once the new IP address becomes dominant, it can be reported to the blacklist to replace the old IP address. Fetching the IP address is quicker than fetching the destination web pages, and comparing two IP addresses is easier than comparing two thumbnails of the destination websites.

7.1.2 Forensic Applications

The research results provide forensic evidence to zero in on the cyber criminals. Currently, we store email message IDs associated with a cluster in the database. We can retrieve associated email subjects and embedded URLs using message ID. The index portion of the message ID also indicates the location of a message in a plain text file, which is useful for accessing the raw message text. The domain name portion of the URLs can be used to find the hosting IP addresses and their network ASN number, location and country code. The database schema and related SQL queries can be found in Appendix A.

In the future, we will create three additional tables: a cluster summary table, an IP table and a subject table. The summary table stores cluster IDs, a short description of each cluster, two time-stamps indicating the start and end times. The IP table contains the IP addresses of a cluster and the active period of each IP. The subject table contains the subject patterns of a cluster and the active period of each pattern. A web interface linking to the back-end database will be developed. Using the interface, spam investigators can specify a time period and the interface will return all clusters during that time period. Then the investigator can choose a cluster he wants to investigate and run the automated query to find the IP addresses, domain names and subject patterns of that cluster. The investigator can do further investigations on the IP addresses and domain names returned using other tools, such as domain WHOIS lookup and domain history. Once solid evidence has established that a group of domain and IP addresses have been used for spam purpose, the investigators can seek the cooperation of law enforcement to shut them down. The investigator can trace the identity of domain owner. Even though the owner information is likely to be phony, the contact email address and the credit card used to pay the domain fee can be traced.

The hosting IP addresses of the spam domains are useful for identifying ISPs which provide bullet-proof hosting service to spammers. If several IP addresses from an ISP are identified as spam hosts, it is likely the other IP addresses in the same network range are also used for spamming. If enough evidence can be gathered, law enforcement personnel can shut down those ISPs so that spammers have to find a new ISP which is willing to do business with them.

If a guilty IP is discovered, the reverse IP lookup can reveal other domains hosted at the same location. Therefore, by monitoring the suspicious IP addresses we can detect new spam domains and disable them even before spammers can put them into use.

7.1.3 Contributions to Data Mining

This research also proposed an algorithm to measure the similarity between two entities, which cannot be represented by vectors. In traditional clustering research, a data point is represented by a vector, containing a series of numerical or binary values. Normal distance functions can calculate the distance between two data points. Statistical data can describe a cluster of data points, such as mean and standard deviation. However, if the data point cannot be represented by a multi-dimensional coordinate, normal distance and statistical functions cannot apply. The similarity between two entities depends on how many other attributes are common to both entities. In our case, the similarity between two spam domains is measured based on the similarity of related email subjects and hosting IP addresses.

Because the data points have no coordinates, traditional centroid-based, distance-based or density-based clustering algorithms cannot be applied to this kind of problem. We can calculate the pair-wise similarity between two entities but cannot define a cluster centroid and shape. Therefore, a graph based clustering algorithm is developed to link entities that are similar to each other and extract connected components as clusters. A bi-connected component algorithm ensures that there are no weak breakpoints in the clusters.

The same approach can cluster other web documents and blog messages which are composed of strings of words and which may contain URLs. Blog spam is another threat to internet security. Cyber criminals may post blog messages to share their experience and tools. Detecting those messages may help uncover the latest developments in criminal technologies.

7.2 Future Work

This research is part of the Spam Data Mining for Law Enforcement project at the University of Alabama at Birmingham. We have established a large database of spam emails on a distributed system, and are receiving a large number of spam from several sources. The focus of the research is to use computing power to process spam emails and extract useful evidence for law enforcement officials to use.

In this research, we used domain names and hosting IP addresses to cluster spam emails, supplemented by email subjects, and produced promising results. In the future, we would like to add more attributes to our clustering. Using more attributes will improve the cluster quality by reducing false relationships among spam emails and discover more relationships among different clusters, and produce a more detailed report of the results. Some of the attributes we are considering include the sender's name and email address and the nameservers of spam domains.

The derived attributes, such as the nameserver, are more useful for forensic analysis than inherent attributes, such as the sender's name and email address, which are usually forged. Sometimes you may receive a spam message from your friend or even yourself. Obviously the sender is someone else who is hiding. Sometimes a spammer

will use a fixed name or a pattern, which might indicate connections between spam emails. An inherent attribute provides helpful supplemental evidence to the clustering of spam. The nameserver of a spam domain will provide additional information about the spamming hosts. Shutting down a spam nameserver will thwart the infrastructure of a spamming network, especially against a single-flux network, a type of FFSN. In a single-flux network, only the web pages are proxied via a flux-bot, which redirects the browser to the real website. In a double-flux network, both the hosting and name servers are proxied via a flux-bot and protected. Therefore for a single-flux network, even though a spam domain's hosting IP addresses are dynamic, its nameservers are still static and can be traced to relate many domains served by the same nameservers. The detection of nameservers may also be useful against botnets that use DNS to boost the search efficiency for C&C servers. The recent shutdown of Waledac botnet is a result of that.

Fetching the spam URLs can find the content of the websites and through that information spam can be classified into categories. We may also identify different spam websites that are hosted at the same servers. However, the retrieval of web pages is time-consuming and some hosts have counter-measures that will block programmed probes. After a certain number of tries, it will always get a time-out response page. Some websites are parked by the time of retrieval. Because of the low successful fetching rate, it is premature to use the fetched web pages in our clustering. But the information is useful for checking the quality of a cluster. In this case, we can sample from the domains in a cluster and fetch their websites. The number of fetches can be limited to a small sample size to avoid being blocked by the hosting servers. If the websites in a cluster

have too much diversity, the cluster is questionable. The fetched web pages can be saved as evidence for forensic purposes.

The algorithm can also be improved in several aspects. First, when calculating the similarity score, we try to find for each item in set A, which is of smaller size, a best match in set B. Because fuzzy matching is involved, multiple items in A may be matched to the same item in B, thus elevating the similarity score. This scenario is not possible in exact matching case where each item can find only one match or no match at all. The best approach is to assign each item in A to an item in B so that the sum of all pair-wise similarities is maximized. A brute-force algorithm to solve the problem has the complexity of $n!/(n-m)!$, where $m=|A|$ and $n=|B|$. Therefore, we need to find an algorithm that solves the problem in less time complexity. A possible approach is to perform the calculation twice, first matching A to B, then B to A and check if the similarity scores are close to each other. If one score is considerably smaller than the other, the above scenario occurs and the smaller score would be taken. This approach does not involve extensive computation because it only needs pair-wise similarity scores which are already computed using the current approach.

Second, current validation is done through human inspection. Without pre-labeled data we cannot measure the precision and accuracy of the results. The false-positives and false-negatives identified by humans sometimes are questionable, and further evidence may reverse the decision. False-negatives are even harder to define because the process requires scanning all the domain names in our dataset and checking their destination websites and registrar information. Some domains may already be dead by the time of validation. In the future, we need to deploy a mechanism that can

automatically fetch validation data and measure the degree of errors in clustering results. Such a mechanism is the basis to build a system with AI. Currently the thresholds used in clustering are pre-set by a human based on observation. A better system should adjust the thresholds based on the accuracy of previous results and the feedback from a forensic expert.

However, human involvement may still be necessary because we are dealing with criminals, not machines. A spammer may decide to change his spamming strategies suddenly if the old strategy is no longer effective or if he senses a threat. Therefore, the old training data may become totally obsolete at some points and a human may need to adjust the system accordingly.

As the size of our spam corpus increases, we are currently changing the "daily clustering" to "hourly clustering," and in the future will run clustering at increasingly smaller time intervals. "Emerging clusters" may then be evaluated on a frequency based on the time interval encountered. In our current research, only IP addresses and subjects are used for tracing clusters. In the future, we need to create a repository for more attributes as described in section 6.1. Some attributes need to be compressed before being stored into the repository, such as extracted patterns from email subject and sender's name.

This research only focuses on the spam domains hosted at static IP addresses. For spam domains that use FFSN, particularly found in phishing spam, when the IP addresses outnumber the domain names, the method is ineffective. The point-of-sale spam websites that sell drugs and sexual enhancement products still use static IP hosting, probably because they are seldom attacked by law enforcement agents, making it unnecessary for

spammers to frequently change the hosting IP addresses. They only need to replace the old domain names with fresh ones so that the blacklist cannot keep up with the refresh rate. Therefore, blocking hosting IP addresses will still be effective against point-of-sale spam websites and associated spam emails. However, legitimate websites may be hosted at the same place. Therefore, further investigation of the identified IP addresses is required before the IP addresses can be blacklisted.

LIST OF REFERENCES

Aggarwal, C. C., Han J., Wang, J. and Yu, P. S. (2003). A framework for clustering evolving data stream. *The 29th International Conference on Very Large Data Bases*. Berlin, Germany.

Anderson, D. S., Fleizach, C., Savage, S., and Voelker, G. M. (2007). Spamscatter: Characterizing internet scam hosting infrastructure. *The 16th USENIX Security Symposium*. Boston, MA.

Aradhey, H. B., Gregory, K. M., and James, A. H. (2005). Image analysis for efficient categorization of image-based spam e-mail, In *Proc. of the 8th International Conference on Document Analysis and Recognition, (ICDAR 2005).* Seoul, Korea.

Baase, S. (1988). *Graphs and digraphs, in Computer Algorithms: Introduction to Design and Analysis*, (*2nd ed.*). Addison-Wesley, Boston, MA.

Barbara, D. (2002). Requirements for clustering data streams. *ACM SIGKDD Explorations Newsletter*, 3(2), 23 – 27.

Bergholz, A., Paass, G., Reichartz, F., Strobel, S., Moens, M-F. and Witten, B. (2008). Detecting Known and New Salting Tricks in Unwanted Emails. *The 5th Conference on Email and Anti-Spam*. Mountain View, CA.

Biggio, B., Fumera, G., Pillai, I. and Roli, F. (2007). Image spam filtering by content obscuring detection. *The 4th Conference on Email and Anti-Spam*. Mountain View, CA.

Blumstein, A., Cohen, J. and Nagin, D. (Eds.) (1978). Incapacitation: Estimating the Effects of Criminal Sanctions on Crime Rates. *National Academy of Sciences*, Washington, DC.

Bollobas, B. (1998). *Modern Graph Theory*. New York City, NY: Springer.

Byun, B., Lee, C. -H., Webb, S., and Pu, C. (2007). A discriminative learning approach to image modeling and spam image identification. *The 4th Conference on Email and Anti-Spam*. Mountain View, CA.

Byun, B., Lee, C. -H., Webb, S., Irani, D. and Pu, C. (2009). An Anti-spam Filter Combination Framework for Text-and-Image Emails through Incremental Learning. *The 6th Conference on Email and Anti-Spam*. Mountain View, CA.

Calais, P. H., Pires, D. E. V., Guedes, D. O., Meira, W. Jr., Hoepers, C. and Steding-Jessen, K. (2008). A Campaign-based Characterization of Spamming Strategies. *The 5th Conference on Email and Anti-Spam*. Mountain View, CA.

Cao, F., Ester, M., Qian, W. and Zhou, A. (2006). Density-Based Clustering over an Evolving Data Stream with Noise. *The 6th SIAM International Conference on Data Mining*. Bethesda, MD.

Claburn, T. (2010). Microsoft decapitates Waledac botnet. *InformationWeek*. Retrieved from http://www.informationweek.com:80/news/hardware/desktop/showArticle.jhtml?articleID=223100747.

Clark, J., Koprinska, I. and Poon, J. (2003). A neural network based approach to automated e-mail classification. In *Proc. of IEEE/WIC International Conference on Web Intelligence*, 13, (17), 702 – 705.

Clayton, R. (2009). How much did shutting down McColo help? *The 6th Conference on Email and Anti-Spam*. Mountain View, CA.

Cockerham, R. (2007). There are 600,426,974,379,824,381,952 ways to spell Viagra. Retrieved from http://cockeyed.com/lessons/viagra/viagra.html.

Cooke, E., Jahanian, F., and McPherson, D. (2005). The zombie roundup: understanding, detecting, and disrupting botnets. *The Steps to Reducing Unwanted Traffic on the Internet Workshop*. Cambridge, MA.

DiBenedetto, S., Massey, D., Papdopoulos, C. and Walsh P. J. (2009). Analyzing the aftermath of the McColo shutdown. *The 9th Annual International Symposium on Applications and the Internet*. Seattle, WA.

Drucker, H., Wu, D. and Vapnik, V.N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10, (5),* 1048 – 1054.

Goebel J. and Holz, T. (2007). Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In *Proc. of USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*. Cambridge, MA.

Grizzard, J., Sharma, V. and Dagon, D. (2007). Peer-to-peer botnets: overview and case study. In *Proc. of USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*. Cambridge, MA.

Gu, G., Perdisci, R., Zhang, J. and Lee, W. (2008). BotMiner: Clustering analysis of network traffic for protocol-and structure –independent botnet detection. In *Proc. of the 17th USENIX Security Symposium (USENIX '08)*, 139-154, Boston, MA.

Gu, G., Porras, P., Yegneswaran, V., Fong, M. and Lee, W. (2007). BotHunter: Detecting malware infection through ids-driven dialog correlation. In *Proc. of the 16th USENIX Security Symposium (Security'07)*. Boston, MA.

Gu, G., Zhang, J. and Lee, W. (2008). BotSniffer: Detecting botnet command and control channels in network traffic. In *Proc. of the 16th Annual Network and Distributed System Security Symposium (NDSS'08)*, San Diego, CA.

Gupta, M. (2008). Flux in Fraud Infrastructures. *The IEEE 22nd Annual Computer Communications Workshop*. Steamboat Springs, CO. Retrieved from http://www.netsec.colostate.edu/ccw08/Slides/gupta-1.ppt.

Gusfield, D. (1997). *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge, UK: Cambridge University Press.

Holz, T., Corecki, C., Rieck, K. and Freiling, F. C. (2008). Measuring and detecting fast-flux service network. In *Proc. of the 16th Annual Network and Distributed System Security Symposium (NDSS'08)*, San Diego, CA.

Holz, T., Steiner, M., Dahl, F., Biersack, E. and Freiling, F. (2008). Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *Proc. of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*. San Francisco, CA.

Jeh, G. and Widom, J. (2002). SimRank: A measure of structural-context similarity. In *Proc. of the 8th ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD2002)*. Edmonton, Alberta, Canada.

Kanich, C., Kreibich, C., Levchenko, K., Enright, B., Voelker, G., Paxson, V. and Savage, S. (2008). Spamalytics: An empirical analysis of spam marketing conversion. *The 15th ACM Conference on Computer and Communication Security (CCS2008).* Alexandria, VA.

Kaplan. D. (2010). Waledac demise imminent after shutdown of domains. *SC Magazine*. Retrieved from http://www.scmagazineus.com/waledac-demise-imminent-after-shutdown-of-domains/article/164535/.

Karasaridis, A., Rexroad, B. and Hoeflin, D. (2007). Widescale botnet detection and characterization. In *Proc. of USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*. Cambridge, MA.

Kirk, P. (2003). Gnutella Protocol. Retrieved from http://rfc-gnutella.sourceforge.net/.

Konte, M., Feamster, N. and Jung, J. (2009). Dynamics of online scam hosting infrastructure, *The 10th Passive and Active Measurement Conference (PAM2009)*. Seoul, South Korea.

Lee, H. and Ng, A. (2005). Spam Deobfuscation using a Hidden Markov Model. *The 2nd Conference on Email and Anti-Spam*. Stanford Univ., CA.

Lee, S., Jeong, I. and Choi, S. (2007). Dynamically weighted Hidden Markov Model for spam deobfuscation. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*, 2523-2529. Hyderabad, India.

Levenshtein, V. I. (1966). Binary codes capable of correcting insertion and reversals. *Soviet Physics - Doklady*, *10*, 707 – 710.

Liu, C and Stamm, S. (2007). Fighting unicode-obfuscated spam. In *Proc. of the anti-phishing working groups 2nd annual eCrime researchers summit (APWG2007)*. 45 – 59. Pittsburgh, PA.

M86 Security. (2008). Sex, Drugs and Software Lead Spam Purchase Growth. Retrieved from http://www.marshal.com/pages/newsitem.asp?article=748.

Maymounkov, P. and Mazières, D. (2002). Kademlia: A peer-to-peer information system based on the XOR metric. In *Proc. of the 1st International Workshop on Peer-to-Peer Systems*, pp. 53-62, Cambridge, MA.

McAfee Avert Labs. (2009). McAfee threats report: first quarter 2009. Retrieved from http://img.en25.com/Web/McAfee/5395rpt_avert_quarterly-threat_0409_v3.pdf.

Mori, T., Esquivel. H., Akella. A., Shimoda, A. and Goto. S. (2009). Understanding the World's Worst Spamming Botnet. Retrieved from ftp://ftp.cs.wisc.edu/pub/techreports/2009/TR1660.pdf.

Passerini, E., Paleari, R., Martignoni, L. and Bruschi, D. (2008). FluXOR: detecting and monitoring Fast-Flux Service Networks. In *Proc. of the 5th GI International Conference on Detection of Intrusions and Malware & Vulnerability Assessment* (*DIMVA 2008*). Paris, France.

Pu, C., and Webb, S. (2006). Observed trends in spam construction techniques: A case study of spam evolution. *The 3rd Conference on Email and Anti-Spam*. Mountain View, CA.

Qi, M., Wang, Y. and Xu, R. (2009). Fighting cybercrime: legislation in China. *International Journal of Electronic Security and Digital Forensics, 2*, (2), 219-227.

Ramachandran, A., Dagon, D. and Feamster, N. (2006). Can DNS-based blacklists keep up with bots. T*he 3rd Conference on Email and Anti-Spam*. Mountain View, CA.

Ramachandran, A. and Feamster, N. (2006). Understanding the network-level behavior of spammers. *The 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. Pisa, Italy.

Ramachandran, A., Feamster, N. and Dagon, D. (2006). Revealing botnet membership using DNSBL counter-intelligence. In *Proc. of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06).* San Jose, CA.

Ramachandran, A., Feamster, N. and Vempala, S. (2007). Filtering spam with behavioral blacklisting. In *Proc. of the fourteenth ACM Conference on Computer and. Communications Security,* Alexandria, VA.

Sahami, M., Dumais S., Heckerman, D. and Horvitz, E. (1998). A Bayesian approach to filtering junk email. *AAAI Workshop on Learning for Text Categorization, AAAI Technical Report WS-98-05*. 55 – 62. Madison, Wisconsin.

Sanpakdee, U., Walairacht, A. and Walairacht, S. (2006). Adaptive spam mail filtering using genetic algorithm. In *Proc. of the 8th International Conference on Advanced Communication Technology*. 441 – 445.

Schoof, R. and Koning, R. (2007). Detecting peer-to-peer botnets. Retrieved from http://staff.science.uva.nl/~delaat/sne-2006-2007/p17/report.pdf.

Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J. and Zhang, C. (2009). An empirical analysis of phishing blacklists. *The 6th Conference on Email and Anti-Spam*. Mountain View, CA.

Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, *16*(1), 30–34.

Soonthornphisaj, N., Chaikulseriwat, K. & Piyanan, T. (2002). Anti-spam filtering: a centroid-based classification approach. *In Proc. of 6th International Conference on Signal Processing, 2,* 1096 – 1099. Innsbruck, Austria.

Spamhaus DBL. (2010). Retrieved from http://www.spamhaus.org/dbl/.

Spamhaus PBL. (2010). Retrieved from http://www.spamhaus.org/pbl/.

Spamhaus ROKSO. (2010). 100 Known Spam Operations responsible for 80% of your spam. Retrieved from http://www.spamhaus.org/rokso/index.lasso.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (*SIGCOMM 01),* 149-160, New York, NY.

St Sauver, J. (2008). Spam, domain names and registrars. *MAAWG 12th General Meeting*. San Francisco, CA. Retrieved from http://www.uoregon.edu/~joe/maawg12/domains-talk.pdf.

Strayer, W. T. , Walsh, R., Livadas, C. and Lapsley, D. (2006). Detecting botnets with tight command and control. In *Proc. of the 31st IEEE Conference on Local Computer Networks (LCN'06)*. Denver, CO.

The Honeynet Project. (2007). Know your enemy: Fast-Flux Service Networks. Retrieved from http://www.honeynet.org/papers/ff/.

Tom, P. (2008). Latent botnet discovery via spam clustering. *The Expanded MIT Spam Conference 2008*. Boston, MA.

Wang, Z., Josephson,W., Lv, Q., Charikar, M. and Li, K. (2007). Filtering image spam with near-duplicate detection. *The 4th Conference on Email and Anti-Spam*. Mountain View, CA.

Webb, S., Caverlee, J. and Pu, C. (2006). Introducing the Webb Spam Corpus: Using email spam to identify web spam automatically. *The 3rd Conference on Email and Anti-Spam*. Mountain View, CA.

Webb, S., Caverlee, J. and Pu, C. (2007). Characterizing Web Spam Using Content and HTTP Session Analysis. *The 4th Conference on Email and Anti-Spam*. Mountain View, CA.

Webb, S., Caverlee, J. and Pu, C. (2008). Predicting web spam with HTTP session information. In *Proc. of the 7th Conference on Information and Knowledge Management (CIKM 2008)*. Napa Valley, CA.

Wei, C., Sprague, A. and Warner, G. (2009). Clustering malware-generated spam emails with a novel fuzzy string matching algorithm. In *Proc. of the 2009 ACM Symposium on Applied Computing*. 889-890. Honolulu, HI.

Wei, C., Sprague, A., Warner, G and Skjellum, A. (2008). Mining spam email to identify common origins for forensic application. In *Proc. of the 2008 ACM Symposium on Applied Computing*. 1433-1437. Fortaleza, Ceara, Brazil.

Wei, C., Sprague, A., Warner, G and Skjellum, A. (2010). Clustering Spam Domains and Destination Websites: Digital Forensics with Data Mining. *Journal of Digital Forensics, Security and Law, 5*, (1).

WikiPedia. (2009). Wildcard DNS Record. Retrieved from http://en.wikipedia.org/wiki/Wildcard_DNS_record.

Zdrnja, B., Brownlee, N. and Wessels, D. (2007). Passive monitoring of DNS anomalies. In *Proc. of the 4th GI International Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2007),* Lucerne, Switzerland.

Zhang, C., Chen, X., Chen, W-B., Yang, L. and Warner, G. (2009). Spam image clustering for identifying common source of unsolicited emails. *International Journal of Digital Crime and Forensics, 1*, (3), 1-20.

Zhang, T., Ramakrishnan R. and Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. *The 1996 ACM SIGMOD International Conference on Management of Data*. Montreal, Canada.

Zhao, W. and Zhang, Z. (2005). An email classification model based on rough set theory. In *Proc. of the 2005 International Conference on Active Media Technology*. 403 – 408. Takamatsu, Kagawa, Japan.

Zhou, A., Cao, F., Qian, W. and Jin, C. (2007). Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems, 15*, (2), 181– 214.

APPENDIX A

SPAM DATABASE DESCRIPTION

Spam Database Description

The spam database is a PostgresSQL relational database. Figure 21 lists the tables that have been used in this research.

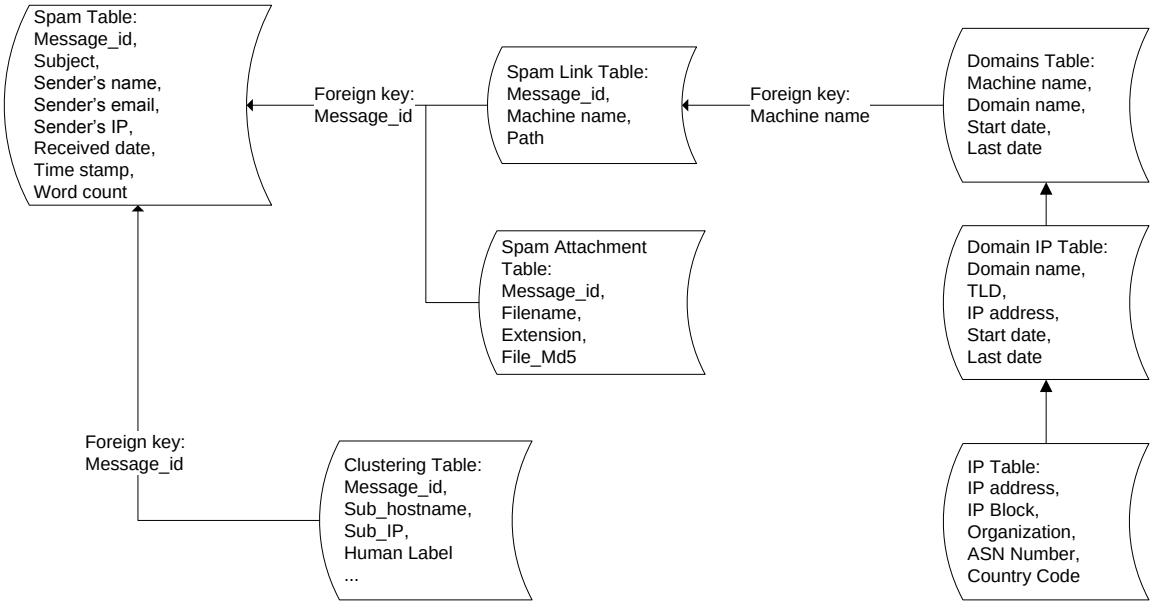

Figure 21: Spam database schemas

The main table "spam" contains attributes that are retrieved directly from the email body. The email text is not stored in the database.

Each email message is assigned a unique message_id, which is in the format of "xxx.09Dec01.1415.1033". The first token "xxx" indicates the source of the spam. The second token "09Dec01" indicates the year, month and day of the spam message, in this case, the date is Dec. 1, 2009. The third token is the timestamp, "1415" meaning 14:15. The last token is the index.

The first three tokens together correspond to the filename, which stores the actual spam messages. All email messages received on Dec. 1, 2009 at 14:15 will be stored in a file named "xxx.09Dec01.1415". The messages are stored sequentially, so the index can be used to find a specific message.

Other attributes stored in the "spam" table are: email subject, sender's name, sender's email address, sender's IP address, the received date, the received timestamp and the number of words in email body.

The spam link table stores the URLs extracted from email messages. The URL is separated into hostname (corresponding to machine name in the table) and path. The table is linked to the main table by message_id.

The spam attachment table stores the information of spam attachment, including attachment filename, file extension and MD5 hash of the file. The table is also linked to the main table by message_id. The real file is stored elsewhere. The filename is the original filename prefixed with the MD5 hash value.

The domains table stores the hostname, domain name, start and end date of the hostname. The hostnames are pulled from the spam link table and de-duplicated. Then domain name portion of the hostname is extracted.

The domain_ip table stores the IP addresses of domain names found in the domains table. A domain name can resolve to several IP addresses and an IP can host many domain names. Therefore, there is a many-to-many relationship between IP and domain in this table.

The IP table stores the network information of IP addresses found in the IP table, including IP block, the organization which owns the IP, ASN number and country code.

The clustering table contains the clustering results. The table is indexed by message_id, which referencing the message_id in the spam main table. The other attributes are cluster labels. The experimental results in Chapter 5 and 6 are stored in sub_ip column. To find all emails belonging to a cluster for a particular hour of a day, we can use the following SQL query:

```
select message_id from clustering where message_id like
'abc.10Jan01.11%' and sub_ip = '116.123.221.91:refillonweb.net';
```

The two parameters are in single quotation. The first one specifies the data source, the date and hour. The second one specifies the cluster label. The query will return all message IDs that meet the criteria. Users can then use message IDs to find out other information of the emails.

The following two queries are used in this research to retrieve useful data for clustering.

1) Acquiring domain and IP addresses:

```
select i.domain, host(i.ip_address), min(i.tld) from
spam_link k, domains d, domain_ip i where k.message_id like ? and
k.machine = d.machine and d.domain_name = i.domain group by
i.domain, i.ip_address order by i.domain, ip_address;
```

2) Acquiring domain and email subjects:

```
"select d.domain_name, s.subject, s.message_id from spam
s, spam_link k, domains d where s.message_id like ? and
s.message_id = k.message_id and k.machine = d.machine group by
d.domain_name, s.subject, s.message_id order by d.domain_name,
s.subject;
```

APPENDIX B

RECURSIVE SEED SELECTION ALGORITHM (PSEUDO CODE)

Recursive Seed Selection Algorithm (Pseudo Code)

```
Input: List of subjects pool S;
Set Q for qualifying subjects
minimum = 1; // Smallest similarity score
While S is not empty
{     seed = S.removeFirst();
      Q.add(seed);
      while (seed is not null)
    {     s1 = seed;
          seed = null;
          minimum = 1;
          while S is not empty
         {      s2 = S.removeFirst();
              If (Similarity(s1, s2) ≥ h)
              {     Q.add(s2);
                    If (Similarity (s1, s2) < min)
                   {     min =  Similarity(s1, s2); // update the minimum similarity score
                         seed = s2;    // set s2 as the new seed    }
              }
            else {  S.add(s2); }  // put s2 back to pool is s2 not similar to the seed
         }
    }
```

APPENDIX C:

BI-CONNECTED COMPONENT ALGORITHM (PSEUDO CODE)

Bi-connected Component Algorithm (Pseudo Code)

Input: G = (V, E), a connected graph (undirected) represented by linked adjacency lists with V={1, 2, …, n}.

Output: Lists of the edges in each biconnected component of G.

```
Array[VertexType] of integer: dfsNumber;
Array[VertexType] of integer: back;
Integer: dfn;
VertexType: v;
Stack: edgeStack;
Procedure Bicomponents (HeadList: adjacencyList, integer: n)
{       For v=1 to n
        { dfsNumber[v] = 0; }
        dfn = 0;
        bicompDFS(1);
}
Procedure bicompDFS(VertexType: v)
{       VertexType: w;
        NodeList: prt;

//process vertex when first encountered.
        dfn++;
        dfsNumber[v] = dfn; back[v] = dfn;
```

```
        ptr = adjacencyList[v];
        w = ptr.nextVertex;
        while (w != null)
        {       If (back[w] < back[v])
                {       push vw on edgeStack;  }
                // else wv was a back edge already examined
                If (dfsNumber[w] == 0)
                {       bicompDFS(w);
                        // now backing up from w to v
                        If (back[w] >= dfsNumber[v])
                        {       // output bicomponent
                                do {  output pop(edgeStack); }
                                while (vw is not popped);
                        }
                        else   // not a bicomponent
                        {       Back[v] = min(back[v], back[w]);  }
                }
                else // w is already in the tree
                {       back[v] = min(dfsNumber[w], back[v]);   }
        // end processing w, proceed to the next vertex
        w = ptr.nextVertex;
        }  // end while loop
}
```