Analyzing Network and Content Characteristics of Spim using Honeypots

Aarjav J. Trivedi
Applied Research
Secure Computing Corporation
atrivedi@securecomputing.com

Paul Q. Judge
Applied Research
Secure Computing Corporation
pjudge@securecomputing.com

Sven Krasser
Applied Research
Secure Computing Corporation
skrasser@securecomputing.com

Abstract

Instant messaging spam (spim), while less widespread than email spam, is a challenging problem which has received little attention in formal research. Spim is harder to study than spam because of the "walled garden" nature of popular instant messaging platforms. We designed and deployed a proxy based IM honeypot with protocol decoding and analyzed content characteristics of spim and network characteristics of hosts sending spim. Our analysis strongly suggests that adversaries make use of botnets and well coordinated command and control mechanisms in sending spim.

Current anti-spim mechanisms rely heavily on content filtering, whitelisting and blacklisting. Our analysis suggests that the same botnets are being employed by spimmers and spammers. Hence network-layer and cross-protocol information sharing between email and IM anti-spam solutions and the use of cross-protocol IP reputation would significantly improve blocking rates. By comparing spim and ham IM data, we also identify several heuristics that can be used to distinguish spim traffic from spam traffic.

1 Motive and Background

Over the past few years, instant messaging (IM) has gone from becoming a tool for communication between friends to a mission critical corporate application at many companies. According to Gartner (a leading market analysis firm), more than 65 percent of all organizations already use instant messaging [1]. Because spim is not as widespread as spam and users share their IM ID less liberally than their email ID, they are more inclined to trust content sent over IM. IM is more real-time in nature than email, guaranteeing an immediate audience to spimmers if they can get through to the recipient. Because of misplaced trust, a spim recipient is more likely to click on a URL without realizing it is not from someone they know. This exacerbates attacks where URLs to mal-ware or phishing sites are sent using the same mechanisms as spim. An example of this is a recent bout of phishing IMs which resulted in a significant number of Yahoo accounts being compromised [2].

Sending spim is harder than sending spam because it requires the spimmer to have a working account on the IM platform unlike email where one can setup their own email server. However new developments like the recently announced interoperability between the Yahoo and MSN messaging platforms means that spimmers now have a much wider audience than before with a single account. Similarly Google Talk supports open Jabber federation, which could allow anyone running a Jabber server to send messages to Google Talk users. With a little bit of effort, it is possible even today to connect Google Talk to AIM, Yahoo and MSN services. All of these could lead to a large increase in spim in the future.

The same reasons that have historically made it harder to spim than spam have also made it harder for re-searchers to study spim. Unlike email, which passes through a receiving server where it can be filtered, instant messages typically pass only through a proprietary platform server. People typically do not store their instant messages, and most IM clients do not have built in forwarding mechanisms, making user reporting of spim harder. Furthermore, if one were to setup an account on an IM platform, proxy it through an instant messaging proxy at the receiving end to log and filter data, and receive spim on it, in most cases the platform, acting as an intermediary between the spimmer and the spim recipient, would not reveal the IP address of the spimmer unless they attempted to

explicitly establish a direct connection as required for e.g. file transfers. Attempting file transfers is atypical for most spim attacks. Current anti-spim mechanisms in IM clients and corporate IM security appliances seem to depend mainly on elementary content filtering, black listing and whitelisting. It is unknown what anti-spim mechanisms are used at the server level by IM platforms. Thus we are faced with a scenario where it is becoming increasingly important to analyze spim and identify new methods to counter it but this has so far been hard.

The rest of the paper is organized as follows. In Section 2 we discuss related work in IM security. In Section 3 we describe the design and setup of our spim honeypot system. In Section 4 we describe the data we collected, identify interesting trends therein, analyze their implications, and discuss the difference between spim and ham IM characteristics. Based on this, we propose heuristics that can be used to identify spim. Section 5 has our concluding remarks including future research possibilities.

2 Related Work

Most popular instant messaging protocols in use today (MSN, AIM, Yahoo) provide whitelisting and blacklisting as options in their client software. AIM also allows user reporting of spim and a rudimentary reputation mechanism for IM IDs which allows users to "warn" an offending IM sender. A large number of warnings result in decreased sending capabilities. Most instant messaging platforms also have built in rate limiting preventing senders from sending a large number of messages in a short period of time. Both blacklisting and whitelisting have well known shortcomings [3,4]. Blacklisting is inefficient in blocking spam in an environment where a large number of senders are available to a spammer. The spammer can abandon and replace blacklisted senders quickly rendering blacklist entries useless. Whitelisting, while useful for users who know everyone they expect messages from, can pose problems in a typical corporate environment where employees may receive messages from customers or partners who's IDs are not known in advance. Rate limiting could be effective against spimmers who send large amounts of spim from a small number of senders. However, our data suggest that they employ a large number of senders sending at low rates.

Formal research in spim is scarce. Liu et al. suggest the use of black/whitelisting, challenge response and content filtering mechanisms [5]. Challenge/response, while a viable anti-spim mechanism, has some well known drawbacks including reduced usability and reduced ability to automate response generation. This is especially relevant when contending with spammers who have a large scale distributed system such as a botnet at their disposal. The common subsequence based filtering approach they suggest needs a spim message to have at least 6 words in common with a previously seen spim message for acceptable false positive rates. As we demonstrate, in practice this assumption is likely to be invalid because spimmers are already injecting randomized markup tokens into spim, even in the middle of a URL, resulting in less than 2 common tokens between spim messages advertising the same URLs in most cases. Furthermore, randomizing parts of the URL itself and registering new "front" URLs is cheap and already widely used by spimmers. These would also cause poor performance in a fingerprint vector based filtering approach. The authors do not specify the lengths of the "short email spam messages" they use to test their Bayesian filtering approach and hence it is hard to predict its performance on real world spim.

Mannan et al. discuss security threats to existing IM networks in [22]. Their analysis includes "Malicious Hyperlinks" which they define as "links to web pages containing malicious content" and a mechanism in the ICQ IM client that allows a user to accept or reject messages containing hyperlinks but they offer no suggestions on how to selectively block spim. They have also implemented an IM key exchange mechanism [23] that would allow "secure communications where authentication, integrity and confidentiality are achieved". However this is unlikely to prevent spim which, even today, comes from authenticated users. Integrity and confidentiality are not directly relevant to the issue of spim. Their findings on (ham) text message rates per user per day in their paper on IM worms [24] are consistent with the rate of ham messages per session presented we observe in section 4.4.

Honeypots are widely used to track and observe malicious network-level behavior [6,7,8]. Similarly the use of honeypots to study spam and spammer behavior has been widely researched [9,10,11]. To the best of our knowledge, honeypots have not been used to study instant messaging spam so far.

3 System Design, Deployment and Security concerns

3.1 Network setup and Security

The intent of a spim honeypot is to appear as if it is an open socks proxy to a spimmer. Spimmers prefer routing through an open proxy in order to conceal their identity. The external firewall should allow in only traffic intended for the socks port (1080) on the proxy. An important concern in choosing what outbound traffic should be allowed from the honeypot to the Internet is making sure that the honeypot is not misused. An open socks proxy is attractive to many

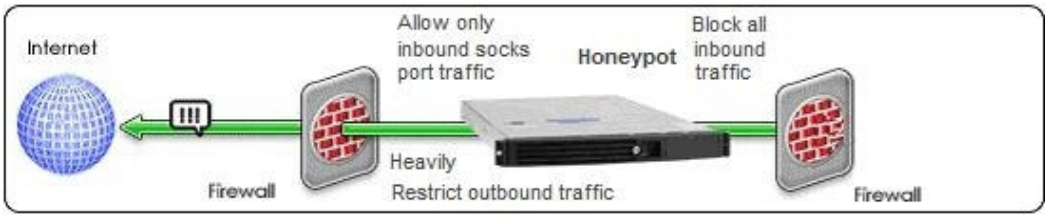

Figure 1.Setting up the spim honeypot

malicious users other than spimmers. For example, email spam or a denial of service attack could be routed through such an open proxy. Hence it would be advisable to block outbound traffic to everything but the IM servers for the IM platforms you intend to proxy and then only to specific ports on these servers. Ports and hostnames for popular instant messaging platforms are well known.

Furthermore, it would be advisable to rate limit the outbound traffic on the external firewall. The decision on what the limit should be is an ethical tradeoff made by all honeypot administrators and is left up to the reader. If a spimmer cannot proxy any traffic through the honeypot to an intended victim, they will not use the honeypot at all. On the other hand, allowing traffic through at a high rate basically amounts to playing into the hands of spimmers. The inbound firewall should block all inbound traffic from the DMZ to the internal network (if any).

3.2 Protocol decoding

All traffic on the honeypot can be logged using a tool such as Wireshark [11] (formerly Ethereal). Wireshark itself has a protocol decoder for popular messaging protocols such as Yahoo, MSN and AIM that can be used to analyze the traffic. However extrapolating trends from a large set of instant messages using Wireshark is hard. While most popular messaging protocols are closed (other than Google Talk, which is largely based on XMPP [12]), information on these is available widely on the Internet [13,14,15]. Using this information and open source libraries available on the Internet [16,17,18] it is possible to extract the necessary information from network level logs. We used an internally developed decoder that allowed us to extract and study spim. We present a very simple decoder mechanism in Appendix I.

4 Analysis of Collected Data

We collected data on our honeypot over a period of two weeks. Over this period, our honeypot registered 228,682 connection attempts from spimmers attempting to send spim. They were able to start 16454 sessions. Across these sessions, they sent 20256 spim messages containing 10542 total URLs and 7386 functional URLs. These messages were sent by 6952 distinct IM usernames to 14249 IM usernames. Each IM username sent fewer than 3 messages on average which seems to indicate a strategy to avoid detection based on per username rates.

During this period we also received connections and messages from individuals who did not appear to be spimmers. These were typically individuals who appeared to be proxying through the honeypot simply for anonymity rather than spimming, as evidenced by the lack of a single URL sent across a multi-message session. We also logged a few hundred messages sent from a single ID to 5 other IDs that did not contain a URL and appeared to be an attempt to overwhelm the victims with a large number of messages. In the interest of focusing on "commercial spim" rather than "targeted personal spim," we have discounted these from our analysis of spim messages

4.1 Content level analysis and trends

The intent of all spam is financial gain and to this end, it usually includes a "call to action." For email spam this is usually a URL. Hence we focused our initial content analysis on the websites being advertised using spim. In the graphs below, we have labeled each website with a unique token instead of identifying it. A list of the actual websites and URLs these tokens represent has not been included for legal reasons.

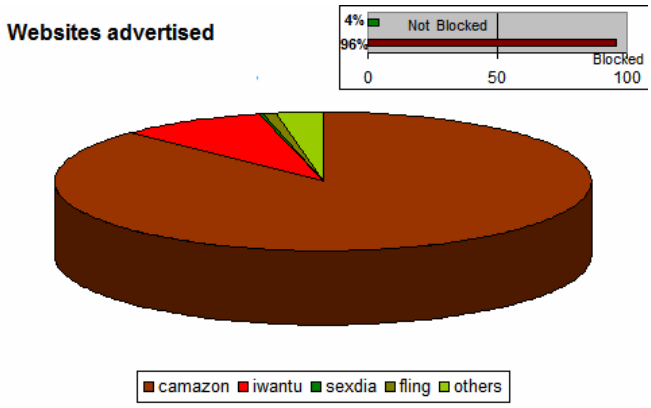

Figure 2.Websites advertised in spim

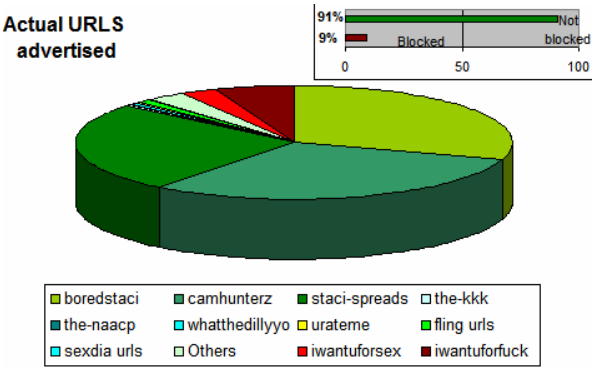

Figure 3.Actual URLs being advertised in spim

Our analysis showed that a very large proportion of the functional URLs being advertised led to a single website (87%). More than 97% of the URLs advertised led to just 4 websites all of which were pornographic in nature. Of these, the 2 with the highest number or URLs advertised were blocked by SURBL. Percentage wise 96% of the spim messages would be blocked by SURBL if the actual website they led to were analyzed. Essentially it looked like spimmers had carried out 4 major campaigns.

The website with the highest number of URLs leading to it (camazon) was never advertised using its actual URL. Instead, 6 "front" URLs were used that led to it. 3 of these front URLs suggested a website with sexual content (boredstaci, stacispreads, camhunterz). 2 suggested racial or political content (the-kkk, the-naacp) and 1 suggested humor or general content (whatthedillyyo). This pattern, where the actual website being advertised was rarely used in spim, was repeated across other campaigns. We then analyzed the actual URLs for presence on SURBL and found that only 9% of these were actually blocked by SURBL. Hence use of SURBL merely as a blacklist on the spim content would have allowed 91% of the spim traffic through.

Analysis of the sending frequency per day of URLs in the 'camazon' campaign revealed that it was likely carried out in two parts, each utilizing 3 URLs. Furthermore the campaigns showed a distinct 'weekly' pattern with the number of spim messages being sent increasing towards the middle of the week and falling towards the weekend. This pattern is not seen in spam email [19].

Analysis of the daily frequency for two other campaigns ('bush/fling' and 'iwantu') revealed that they had a similar sending pattern but did not match with the sending pattern of the 'camazon' URLs or of each other. The first URL was sent on the first day the honeypot was up. However for the entire first week only 0 to 7 URLs of each type were sent out. These numbers reached close to 60 exactly one week after the honeypot was up and more than doubled from the next day forward. The two other campaigns also showed the same weekly sending pattern as the 'camazon' campaign with the number of spim messages sent increasing towards the middle of the week.

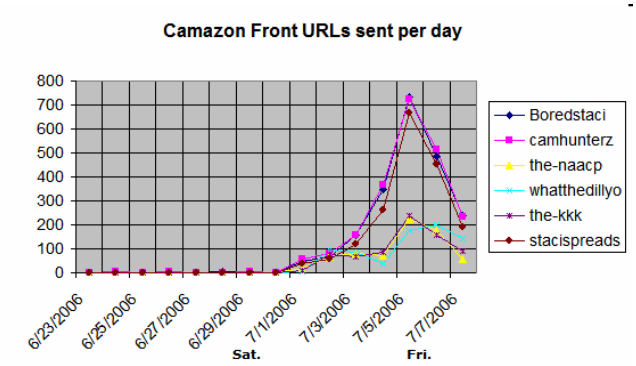

Figure 4.Daily frequency of Camazon front URLs

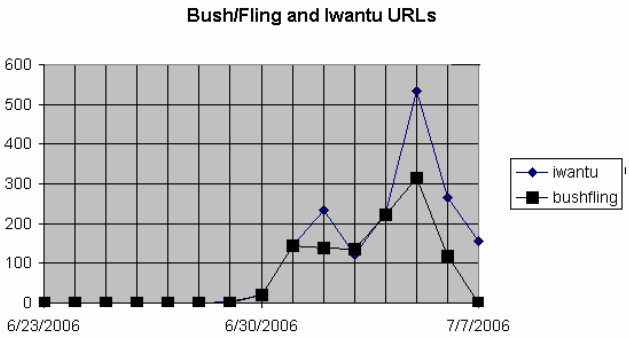

Figure 5.Daily frequency for two other campaigns

The similar rate of sending across different campaigns suggested the possibility that these campaigns were being carried out by the same individuals. Analyzing the user IDs of spimmers confirmed this. They all followed a pattern of a female first name followed by an underscore, two random letters and 4 random integers. This suggested that either the same individuals were carrying out different campaigns or at the very least the same spimming software was being used. Taken together with similar sending rates and network level analysis covered in Section 4.2, we found strong evidence of a single controller behind all campaigns.

The average length of spim messages sent was 144.04 characters. The average number of tokens, separated by space, in a spim message was 10.30. This put the length of an average token at 13.98 characters. Considering only those spim messages that contained a URL, the average length of spim messages containing a URL was 220.81 characters. The average number of tokens, separated by space, in a spim message containing a URL was 16.577. This put the length of an average token for spim messages containing a URL at 13.32 characters. The average length of an English word is considered 4.5 letters [20]. The discrepancy in these

figures is explained by the presence of randomized markup text present in all messages. An example of this with two spim messages advertising the exact same URL and how they are rendered in a messenger client is shown in figure 6. As evident, the bottom two messages show up as the exact same English content in an IM client barring a randomized 3 letter token at the end but are very different in actual text due to insertion of different randomized markup tokens.

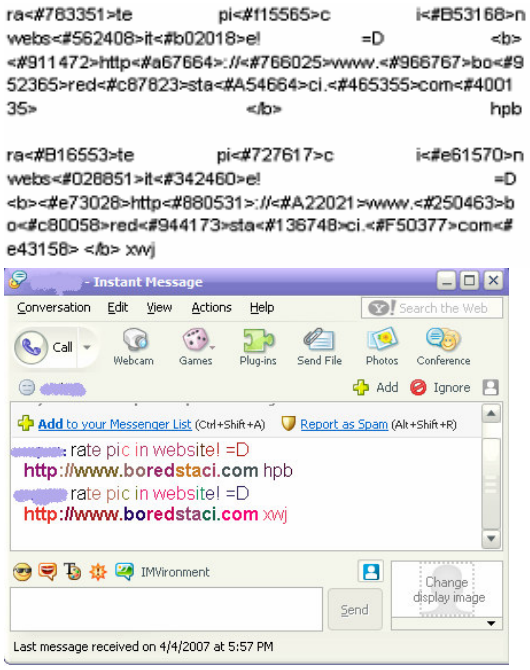

Figure 6.Actual text in spim versus rendering in client

Taken at the raw level, this insertion of random markup would reduce the accuracy of any content filtering solution that didn't explicitly remove these tokens. The random 3 letter token at the end seems to be intended for the same purpose – avoiding filtering at both raw (server) and representation (client) levels.

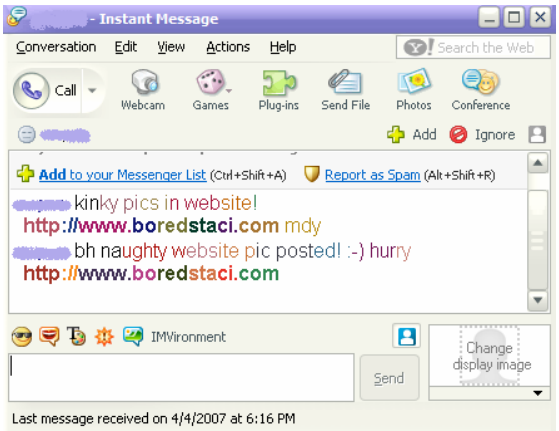

Figure 7.Different surrounding text advertising the same URL

Figure 7 depicts two other messages advertising the same URL but with different English text surrounding each, also possibly aimed at defeating content filtering.

4.2 Network-level analysis and trends

Over a period of two weeks, a total of 3148 unique IP addresses established 228682 connections to our IM honeypot. Of these 2081 unique IP addresses actually tried to send spim. Table 1 shows the distribution of countries where these IP addresses originate. Only the top 12 out of 45 total countries have been individually shown. Table 2 shows the same distribution for IP addresses of email spammers.

As with spam, USA remains the country with the single largest number of IPs sending spim followed by Iran, Germany and France. 3 of the top 5 countries (USA, Germany and France) are common between the spam and spim lists. This, along with the large number of nations where spim advertising a comparatively smaller set of websites originates, supports our thesis that botnets are being used to send spim.

To identify whether the same zombies were being used to send spam and spim, we checked the email reputation of IP addresses sending spim in our Trusted Source [21] for email database. Trusted Source operates on a wide range of public and proprietary data sources. The former includes public information obtained from DNS records, WHOIS data and real-time blacklists (RBLs). The proprietary data is gathered by several thousand IronMail™ anti-spam appliances.

We found that over 29% of IP addresses sending spim had either a 'spam' or a 'suspicious' reputation classification in Trusted Source indicating that they were either being actively used to send email spam or had several behavioral characteristics of an email spam sender. An 'unverified' classification indicates that the IP address has shown neither of these behaviors.

Table 1. Country wise distribution - Spim IP addresses

Country	Percentage
USA	23.55%
Others (33)	14.18%
IRN	13.74%
DEU	8.99%
FRA	7.50%
JOR	7.02%
TWN	5.05%
ITA	4.76%
IND	3.99%
AUS	3.27%
ESP	2.93%
VNM	2.55%
GBR	2.50%

Table 2.Country wise distribution – Spam IP addresses

Country	Percentage
USA	19.08%
CHN	14.56%
KOR	9.61%
DEU	5.99%
FRA	5.69%
BRA	5.56%
JPN	3.70%
GBR	3.13%
ESP	2.96%

Table 3.Email reputation of IP addresses sending spim

Reputation	Percentage	No. of IPs
Unverified	70.39%	1465
Suspicious	13.26%	276
Spam	16.34%	340

4.3 Aggregating network and content level analysis

To find whether the same IPs were being used across separate campaigns, we analyzed IPs used to advertise the Camazon front URLs in the two campaigns shown in Appendix II. We found that two IP addresses were common across all front URLs advertised. Further, each individual campaign had more common IP addresses resulting in higher correlation in daily frequency. This trend was present across all campaigns in our data set.

4.4 Comparing Spim and Ham IM Heuristics

To identify heuristics that can be used to counter spim, we compared data collected from an internal instant messaging proxy deployed at Secure Computing with data from the spim honeypot. We found that for each of the heuristics depicted in Table 4, spim differs from spam by multiple orders of magnitude. By monitoring IP addresses and/or IM users, at the server or proxy level, who exhibit behavior matching the spim heuristics presented here consistently over a certain period of time, it would be possible to identify and block spimmers. The URLs/Message measure, which is almost 100 times larger for spimmers than for normal IM users, the Bytes/session and Messages/Session measures are likely to be quick and effective heuristics for spimmer identification.

Table 4.Spim versus Ham IM heuristics

Heuristic	Spim	Ham IM
Bytes/Session	168	1508.63
URLs/Message	0.68	0.0066
Messages/Session	1.2351	41.35
Avg. Length of messages in characters	144.04	35.45
Avg. number of tokens per message	10.30	5.67
Avg. length of a token in characters	13.98	6.25

5 Concluding Remarks

Our analysis suggests that spim is being sent using widely distributed botnets. Spimmers have already developed tactics to circumvent anti-spim measures that are likely being deployed by IM platforms and organizations. These tactics include the use of unsophisticated but cheap means like using front URLs to sophisticated techniques like randomization of spim content and URL obfuscation. Hence the use of URL blacklists and dictionaries would likely not be an effective measure against spim. Our research suggests that sharing data across protocols at both the content level (i.e. sharing spam URLs advertised in email with anti-spim mechanisms and vice-versa) and at the network level (sharing IP reputation across anti-spam and anti-spim mechanisms) would substantially benefit anti-spam efforts across protocols. Sharing information across protocols about IP addresses which are caught sending spam over one protocol, allows traffic from them to be blocked across several protocols reducing unwanted traffic on the Internet by several multiples.

Our comparison of spim and ham IM characteristics suggests that the use of behavioral heuristics to separate spam from spim is likely to be an effective measure to identify IDs and IP addresses involved in spimming at both for IM platforms and organizational anti-spim mechanisms.

For future research we intend to analyze the performance of email anti-spam algorithms on spim content and how they can be tuned to maximize performance.

Acknowledgements

The authors thank Dmitri Alperovitch for his thoughts and comments.

References

[1] MacDonald, Neil. Recommendations for Infrastructure Protection, 2006. http://mediaproducts.gartner.com/gc/webletter/qwest/issue4/gartner1.htm

[2] InformationWeek. New IM Phishing Attacks Unleashed on Yahoo. Nov. 2005 http://www.informationweek.com/story/showArticle.jhtml?articleID=173601765

[3] Graham, Paul. A Plan for Spam. August 2002. http://www.paulgraham.com/spam.html

[4] Graham, Paul. Filters vs. Blacklists September 2002. http://www.paulgraham.com/falsepositives.html

[5] Zhijun, Liu., Weili, Lin., Na, Li., Lee, D. Detecting and filtering instant messaging spam - a global and personalized approach. 1st IEEE ICNP Workshop on Secure Network Protocols, 2005. pp. 1 -24

[6] Krasser, Grizzard, Owen, Levine. The Use of Honeynets to Increase Computer Network Security and User Awareness http://www.juliangrizzard.com/pubs/2005_krasser_jse.pdf

[7] Lance Spitzner. "Honeyd," Honeypots—Tracking Hackers, Addison Wesley, 2002, pp. 141-166.

[8] The Honeynet Project, "Know Your Enemy: Sebek," http://honeynet.org/papers/sebek.pdf.

[9] N, Provos. A virtual honeypot framework. Proceedings of the 13th USENIX Symposium. 2004. pp. 1-14

[10]Andreolini., Bulgarelli, Colajanni, Mazzoni. HoneySpam: Honeypots fighting spam at the source. Proceedings of the International Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'05). July 2005. pp. 77-83

[11] Schryen, G. An e-mail honeypot addressing spammers' behavior in collecting and applying addresses. Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop, June 2005. pp. 37-41

[12] P Saint-Andre , Ed. RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, 2004

[13] Yahoo Messenger Protocol v-9. http://libyahoo2.sourceforge.net/ymsg-9.txt

[14] MSN Messenger Protocol. http://www.hypothetic.org/docs/msn/

[15] Fritzler, Adam. Unofficial AIM MSN Protocol Specification. http://www.oilcan.org/oscar/

[16] libyahoo2 – A C library for Yahoo Messenger. http://libyahoo2.sourceforge.net/

[17] libmsn a C++ library for Microsoft's MSN Messenger service. http://libmsn.bdash.net.nz/

[18] libpurple – Core GAIM library. http://developer.pidgin.im/wiki/WhatIsLibpurple

[19] Gomes, LH., Cazita, C., Almeida, JM., Almeida, V., Meira, W. Characterizing a spam traffic. Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 2004, pp. 356-369

[20] Shannon, C.E. Prediction and Entropy of Printed English. September 1951. Bell Systems Technical Journal, 30. pp. 50-64.

[21] TrustedSource, http://trustedsource.org

[22] Mannan, Mohammad. and van Oorschot, Paul. Secure public Instant Messaging: A survey. Proceedings of Privacy, Security and Trust, 2004. http://citeseer.ist.psu.edu/mannan04secure.html

[23] Mannan, Mohammad. and van Oorschot, Paul. A Protocol for Secure Public Instant Messaging. Financial Cryptography 2006. pp. 20-35

[24] Mannan, Mohammad. and van Oorschot, Paul. A Proceedings of the ACM workshop on Rapid malcode, 2005. pp 2-11.

Appendix I: Building a sample instant messaging protocol decoder

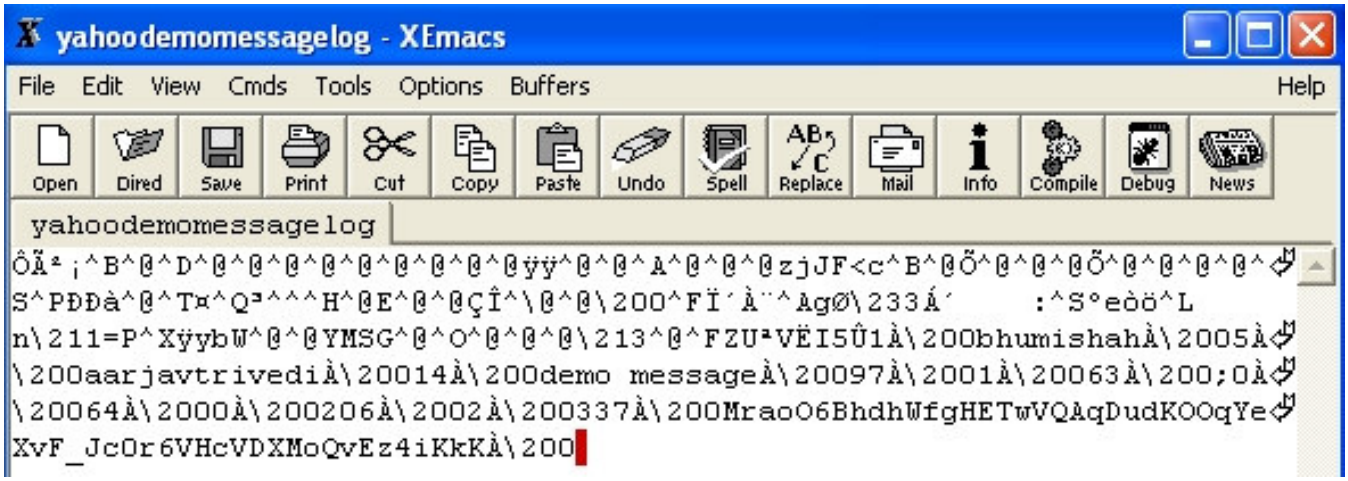

Figure: Wireshark log of a single Yahoo message as seen in Emacs

The figure above depicts the Wireshark log of a single Yahoo message as seen in Emacs. It is easy to write a simple program to extract information about the sender, recipient and the actual text of this message based on information about the Yahoo messenger protocol [13].

For the Yahoo messenger protocol the 0xc080 byte sequence is a separator. In Emacs, this is rendered as Á\200. Yahoo also uses key value pairs where each numeric key denotes a specific variable and is followed by the value of that variable. Three of these variables are:

(0x31) 1: active (sender) id
(0x35) 5: recipient id
(0x3134) 14: message to send

Hence in the message above, 1 is followed by the separator Á\200 followed by 'bhumishah' which is the IM id of the sender, followed by the separator, then '5', denoting that the recipient ID follows, followed by the separator and 'aarjavtrivedi' which is the recipient ID. This is again followed by the separator, then '14', denoting that the actual message follows, the separator and the string "demo message" which was the actual message sent here.

Using the above information, it should be easy to use Wireshark to log all messages, select and save only messages belonging to the Yahoo protocol and write a program to extract the sender, recipient and actual spim message from the message capture files.

Note: In the analysis above, the IM IDs have been altered to preserve privacy.

Appendix II Aggregating Network and Content Analysis

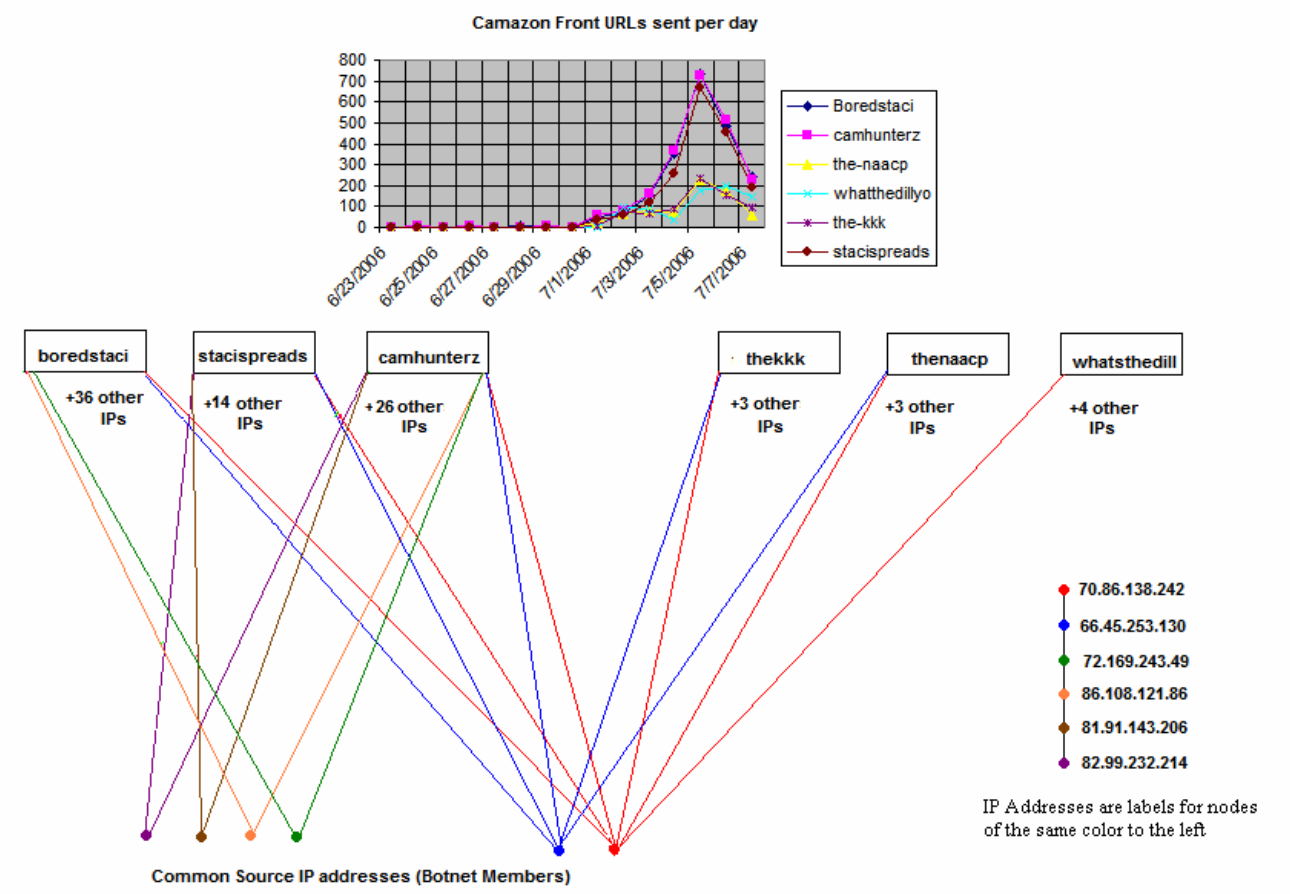

Figure: Correlating common IP addresses across campaigns with their daily frequency across two weeks

The above figure depicts two campaigns carried out by spimmers, one using front URLs that suggest sexual content and the other using front URLs that suggest racial, political or general content. Two IP addresses are common across all 6 URLs advertised across both campaigns. This results in similar but not exactly same shapes in the two aggregate daily frequency graphs, one for each campaign, across two weeks evident in the figure. Further each campaign has an even higher number of common IPs used to advertise the URLs for that campaign. This results in the aforementioned two aggregate curves for each campaign, but each with 3 different curves for each individual URL in that campaign with an almost identical daily frequency.