Discovering Campaign Based Spamming Strategies

Mark David Spadafora

October 2009

University of Auckland

Department of Computer Science

Abstract

This paper introduces the methodologies presently used for the identification of spam email campaigns and the way in which these identified campaigns are used to ascertain specific strategies used in spam delivery. To understand how a spammer operates, it is not enough to simply look at the sum of all of spam messages, to establish a wider understanding, spam messages must be first grouped into campaigns and these campaigns studied as a subset to expose inherent characteristics among spam orchestration operations.

Introduction

Spam email messages are an everyday nuisance for all users of the email system, there are several brute force approaches that detect spam at endpoints and remove them before they get to an inbox, such as filtering systems on a mail daemon using Bayesian statistical methods are is done in the open source SpamAssassin plug-in for Unix based mail servers. These methods are an added afterthought functionality to an email architecture that was not designed to deal with spam messages. These systems can become very clunky and memory intensive on high volume email servers. It becomes apparent through observation of the inefficiency of this method of thwarting spam messages that there is a need to understand spam delivery systems as a whole in order to better defend against spam messages.

Classification of spam campaigns is an important step towards finding specific strategies used by spammers, this is because when looking at all spam messages as a whole we cannot ascertain any characteristics that may point to strategies being used, but by grouping them into campaign we can yield a new measurement to look at which can then be used to further narrow down specific traits in spam campaigns to finally identify spamming strategies. (Calais, et al.)

Two different spam campaign strategy classification methodologies are compared and contrasted, both of which have been implemented and tested in real world spamming situations. We will focus on the methods used to acquire the spam messages and classify them into different campaign subsets and then the further identification of spamming strategies that can be derived from these. The two articles are "Spamcraft: An inside look at Spam Campaign Orchestration" which will be referred to as Spamcraft, and "A Campaign Based Characterization of Spamming Strategies" which will be referred to as CCSS. Upon examination of the two articles, it is clear that the entire process from collection to deriving strategies can be easily labelled into three different stages, Data Collection, Campaign Identification and finally Strategy Identification. There are several points of differences between the two methodologies and to this end, we will be comparing and contrasting the approaches used by the researchers in terms of these three phases.

Spam Campaign Definition

A spam campaign is defined as a set of spam messages that have the same goal, such as advertising a specific brand or product; and that also use the same obfuscation strategy among this set of messages. (Calais, et al.) Obfuscation strategies can include techniques from simply using templates and inserting key words related to the campaign into the templates to attain originality, using a combination of text written into images, to more diverse delivery systems where the spammer will send from domains of many country codes to try and reduce the probability of any pattern being detected. In Spamcraft it is identified that the term 'spam campaign' is a very loose generalisation and in order to further define spam campaigns, they introduce three additional levels of abstraction, classes, types and instances. (Kreibich, et al., 2009)

- Classes

  Intended purpose of the spam eg - phishing, pharmaceutical offers, stock scam.
- Types

  Sets of spam campaigns that have messages that share content which can be matched directly, such as verbatim text.
- Instances

  Sets of spam campaigns that are identical but run across several time periods, delimited by 24 hour time slots.

Although not described verbatim, the CCSS methodology also takes these into consideration, as is evident when looking at the way in which their resultant data is arranged.

Data Collection

It is important to note that the methodologies described in the two approaches do not only differ in relation to their implementation, but also the testing environment, with CCSS being conducted in South America, and Spamcraft being in the United States, as well as CCSS being conducted on a Digital Subscriber Link (Calais, et al.), and the connection for Spamcraft is left unknown, possibly as it is using the same network as their institution. There are inherent differences in available network infrastructure as well as social factors pertaining to the differences in utilisation and congestion of the internet in these countries which have not been addressed.

The data collection methods presented in the two articles are different in their operational scale. With Spamcraft, the study was designed to collect data from a specific botnet, The 'Storm' botnet, whereas the study in CCSS was interested in looking at any spam message abuse that was sent through their system. They are similar in that they both use the idea of a Honeypot, where an array of sensors are operated. In the case of CCSS, they use what is called a 'low-interaction Honeypot' (Calais, et al.) that only identifies with services that a typical spam distribution system would use. The use of a Honeypot works as they do not publically announce themselves to the wider internet by, for example hosting a

website or any other particular service. Therefore any contact with a Honeypot machine can be considered suspicious. The sensors are set up to act as open proxies and open relays. An open proxy is a system that allows any IP address to establish a connection to any destination IP address through itself.

An open relay is an improperly configured SMTP (Simple Mail Transfer Protocol) that allows an unauthorised users to anonymously send mail through the system, this vulnerability occurs as the mail server does not enforce any kind of access control mechanism, i.e. there is no guard to decide if a user can use a system and anybody can directly access it anonymously. (Lampson, 2004) In a correctly configured SMTP server, mail would only be allowed to be sent from users who have previously been identified as users of the system that are authorised to use email services.

The above description of a Honeypot refers to how the data was collected for in CCSS, with Spamcraft however, the Honeypot was custom engineered to specifically collect data from the 'Storm' botnet. In Spamcraft their Honeypot equivalent is called a "Command and Control Rewriter". In addition to this procedure they also used a "Command and Control Crawler" which actively requests a spam workload via the Storm's command and control overnet. The C&C Rewriter also has the ability to inject tracked email addresses to have spam delivered to, this yields another measureable element. (Kreibich, et al., 2009)

The data collection phase in CCSS is conducted for 15 months with their Honeypot architecture, with Spamcraft, the C&C Rewriter is in operation for a total of 2 months, the first of these two months it was collecting data during a passive phase and during second month, collecting data during an active injection of emails phase. Spamcraft's C&C Crawler collected data 10 days short of a year.

The evident differences in the time periods can be rationalised by considering that the experimentation conducted in Spamcraft focused specifically on the storm botnet, conversely CCSS's was conducted on a much broader scale, therefore a larger amount of data is was to accurately differentiate campaigns.

An important consideration made in CCSS is that the spammers do not simply send messages off blindly and assume they are successful, they randomly send out 'probe'

messages to ascertain if a particular path for spam delivery is successful, a spammer may identify any failure of a probe message as an infiltration of their spam delivery network - if it is or not, and therefore cease operation through this path or even send messages that do not represent the actual state of their system, this would clearly undermine the validity of any results gathered. To avoid detection of their effectively null routing setup, CCSS designed their system to identify such probe messages and treat these differently forwarding these as a normal proxy or open relay would. (Calais, et al.) The implications of any detectable activity are not considered in Spamcraft.

In Spamcraft, they do not actually block any communication made through their system, therefore it is, by design more resilient to the abovementioned detection by probing. There are other areas in which their system may be susceptible to detection that are not considered. For example, their system injects email addresses to track email usage. In Spamcraft, the injection of email addresses all come from the same machines that are all using the same IP subnet. Further, all of these injected emails - if examined more closely - are routed to the same mail and DNS servers, and presumably, though not mentioned, - perhaps intentionally - the same IP subnet as that of the infiltration engine. This clearly gives an embarrassing amount of data that could be easily used to identify similarities between injected email addresses.

The spam distribution system is heavily dependent on the fact that it can communicate very easily with new nodes to the distribution system, which means that for a spam master to implement any kind of chain of trust (Lampson, 2004), would be impractical, simply because of the dynamic nature of the distribution system and the fact that the spam master does not essentially control the major parts of the system, it is in this lack of protection that allows both research teams to collect their data.

The models for security defined by Lampson if taken very loosely can be applied to these weaknesses in the spam distribution system, however it is more important to recognise here that Lampson's models are not designed to be directly applicable to such a wide and dynamic distributed network architecture, as is present in spamming networks. So any application of his models to the spamming architecture by extrapolation would undermine the basic security concepts present in the model.

With the collection methods described, both methodologies are actually gaining unauthorised access to the spam masters' distribution system, this opens an ethical question as to the way in which the research is conducted, "is it ethically correct for the researchers to infiltrate the spammer's system in this way and allow the system to falsely assume that mail was delivered?"

This is in clear violation of commandment number two of the "Ten Commandments of Computer Ethics", "Thou shalt not interfere with other people's computer work" As well as violating number three, "Thou shalt not snoop around in other people's computer files" (Barquin, 1992). It could be argued that it is indeed fine as the system being 'snooped' upon is already in violation of other rules. Though this is not the place to discuss such ethics, it is none the less an important issue that has been overlooked by both set of researchers.

Campaign Identification

The approach taken in Spamcraft to identify campaigns is very naive when compared to the sophisticated approach taken in CCSS to identify campaigns.

To identify campaigns, Spamcraft uses the recorded dataset collected from the year long operation of the C&C Crawler. To do this they iterate through the collected spam messages and tokenise them so that the token string can be used to identify characteristics, the characteristics are then grouped together into the different predefined campaign types. It is considered that campaigns may run repeatedly after long periods of inactivity, to identify repeated campaigns, they delineate by dividing into 24 hour time slots where if a campaign is detected in differing time slots, they can be identified as multiple instances of the same campaign. (Kreibich, et al., 2009)

The approach taken in CCSS was much more sophisticated in that they start out by identifying that there are several characteristics that need to be addressed when identifying campaigns, they emphasise the need to not only look at current known obfuscation strategies, but to also design a system that allows future obfuscation strategies to become identifiable. There is detail given pertaining to the assumption regarding the way in which

spammers try to create obfuscated spam messages, all spam messages are created by tools which use the same obfuscation strategies for any given campaign, and make the assumption that a particular instance of a campaign does not run for a period of longer than 24 hours, during this instance it is expected that the obfuscation strategy used in the manufacturing of each spam message in the set of spam messages for that instance is constant. (Calais, et al.)

The data used for their testing is taken from the 15 month operation of their Honeypot proxies. The actual categorisation takes place using a two tiered dissemination of the collected data. The approach identifies frequent patterns and recognises invariant characteristics of the spam messages and further organises them into a hierarchical tree structure. Initially the message is analysed and key characteristics from the message are extracted, specifically: language, textual layout, type of message (HTML, Image, plain text) URL and Subject. Each of these characteristics are derived through different procedures, for example, the 'language used' is deduced by the use of n-grams to detect natural language, for example in the English language, there are rules that show up frequently in n-grams, such as "I before E, except after C" as well as vowel placements etcetera. This approach is very similar to the approach taken in Author Source code classification, where n-grams are used to identify an authors source code using a technique to not only identify natural language but also structure in programming language, this structural application of the n-gram could also potentially be applied to a spam feature identifier to identify message templates. (G. Frantzeskou, 2008)

The inverse of detection can also be accomplished with n-grams, as shown in “dynamic k-gram based software birthmarks” where n-grams are used to hide birthmarks within programs, which are only identifiable if the identifier has knowledge of where to look (Y. Bai, 2008), this technique, if adopted by spammers could possibly make it more difficult to extract features from spam messages. It is to this kind of evolution that the FP-Tree as used by CCSS is resilient. There are several other techniques used to extract each of the aforementioned characteristics that are explained in more detail in the CCSS paper.

Once the characteristics of a message have been identified, they are used in the second step of the campaign identification process, where an FP-Tree (Frequent Pattern

Tree) is constructed to represent the characteristics extracted from the messages. Each node in the tree represents a characteristic of a message mutual to that node's sub tree, therefore messages that share characteristics will traverse the tree in the same manner. The nodes directly off the root node represent the head of several trees that have no commonality between them. A spam campaign can be identified using this method, as when a particular obfuscation strategy is in place on a particular campaign, it becomes visibly evident as sudden increase in the number of child nodes off a particular branch of the tree.

The amount of analysis performed on collected data by the two papers are vastly different. Spamcraft starts off with a very elaborate data collection procedure and CCSS is comparatively less complex. However, during the campaign identification phase, CCSS uses several different tested methods to extract features from the data collected and produce new dimensions for analysis, which when organised into an FP-Tree, produce detailed campaign separation. Contrary to this, the analysis done in Spamcraft is less sophisticated, even though it starts out with a more comprehensive data set. That being said, it must be reiterated that the Spamcraft system was focusing specifically on the Storm Botnet. It would be interesting to take the data collection aspects from Spamcraft and the analysis techniques from CCSS to produce a more complete set of campaigns to move on to the Strategy identification phase with.

Strategy Identification

The strategy identification is less clearly defined as that of the Data collection and Campaign Identification phases, part of the analysis of the strategy done in CCSS is conducted during the Campaign Identification phase. The type of strategies identified by the two approaches are quite different. In Spamcraft there is a heavier focus onto studying the composition of the spam message itself, and the obfuscation strategies used to construct these spam messages.. With CCSS, there is more emphasis on discovering network strategy and particular areas of networks that are abused in a spam distribution system.

Spamcraft identifies several different 'evasive manoeuvres' derived from their results that a spammer uses to obfuscate their messages, briefly described below:

- **Dictionaries** - entries from a dictionary are added to spam messages to introduce variety.
- **Template diversity** - work along with a dictionary to provide an added level of overall diversity
- **Header diversity** - Email headers are chopped up into user agent, mail transfer agent and received by - this yielded 11 permutations.
- **Domain Diversity** - Spammers use hundreds of different domains to provide domain diversity when providing links to products in their spam messages.

(Kreibich, et al., 2009)

CCSS identify several different network level patterns that are evident through the use of the system they set up. Below is a brief summary of the important network characteristics that their study found:

- **Spammers abuse proxies and mail relays differently** - there is a preference for spammers to use proxies rather than mail relays, it is suggested that this preference can be explained by their need to disguise the true origin of the spam.

- **Proxies and relays may be abused in a single campaign** - From the campaigns identified, it is found that 90% of the campaigns they identified do not use open relays at all, whereas 10% use a mixture of proxies and open relays. (but none found using exclusively relays)
- **Spammers Chain Proxies** - spammers chain several proxies together before connecting to open relays and SMTP servers to avoid discovery of the origin.
- **Probe Messages** - Probe messages were seen to originate from the same country as the country codes suggest.

(Calais, et al.)

Both of the strategy identification methods have value to understanding aspects of spamming strategy. Application of the Spamcraft methodology presents more information on spam message composition which can be used in turn to develop better methods to block spam at end points. With CCSS, a clear picture of the ways in which spammers abuse network resources is developed, which will, in turn, help to prevent network level abuse. Stopping the network level abuse would ultimately get at the crux of the problem as it will allow detection of the abuse of network resources and allow it to be dealt before copious amounts of spam are delivered. It is inevitable that some spam would get through regardless, so, although the message composition understanding may seem redundant in the face of actually bringing down the systems which distribute the spam, it is still important. Eventually, it would be more beneficial to amalgamate the two approaches taken.

Conclusion

From the discussion presented in this paper, the approaches taken, Spamcraft and CCSS, have been compared and contrasted against each other to explore the advantages and disadvantages of their methods for the identification of spamming strategies. Three points of comparison between the papers were used to bridge the divide between the approaches, Data Collection, Campaign Identification and Strategy Identification. From looking at these phases, it is seen that there are several concerns pertaining to the way in which data is collected. There are ethical considerations that have been ignored, as well as the way in

which their data collection procedures may be detected. It has been identified that neither of the methodologies taken provide a picture that encompasses the overall state of spamming strategies, with CCSS focusing on the Network Level side of spam distribution and Spamcraft focusing on spam message composition. Both of these add to different aspects on the war on spam, Spamcraft helping to understand End-to-End distribution, and CCSS, discovering network level abuse. For a view that truly encompasses spamming strategies, a combination of the two approaches will ultimately be required.

Works Cited

Barquin, R. C. (1992). In Pursuit of a 'Ten Commandments' for Computer Ethics. *Computer Ethics Institute* .

Calais, P. H., Pires, D. E., Guedes, D. O., Meira, W., Hoepers, C., Steding-Jessen, K., et al. (n.d.). A Campaign-Based Characterization of Spamming Strategies.

G. Frantzeskou, S. M. (2008). Examining the significance of high-level programming features in source code author classification. *Journal of Systems and Software* , 447-460.

Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G. M., Paxson, V., et al. (2009). Spamcraft: An Inside Look At Spam Campaign Orchestration. *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET '09).*

Lampson, B. W. (2004, June). Computer Security in the Real World. *IEEE Computer Society* , 37-46.

Y. Bai, X. S. (2008). Dynamic k-gram based software birthmark. *19th Australian Conference on Software Engineering (ASWEC 2008)*, (pp. 644-649).