Combating Spamming Mobile Botnets through Bayesian Spam Filtering

Ickin Vural, H.S Venter
Department of Computer Science
University of Pretoria
email: Ickin@tuks.co.za,hventer.cs.up.ac.za

Abstract- Malicious software (malware) infects large numbers of mobile devices. Once infected these mobile devices may be involved in many kinds of online criminal activity, including identity theft, unsolicited commercial SMS messages, scams and massive coordinated attacks. Until recently, mobile networks have been relatively isolated from the Internet, so there has been little need to protect them against Botnets. Mobile networks are now well integrated with the internet, so threats on the internet, such as Botnets, have started to migrate to mobile networks. This paper studies the potential threat of Botnets based on mobile networks, and proposes the use of Bayesian spam filtering techniques to detect Spamming Botnets. We then simulate mobile Bot detection by detecting mobile spam originating from a mobile device.

Index Terms—Botnet, mobile, malware, Bayesian spam filtering

I. INTRODUCTION

The field of computer security, which for many years focused on paradigms such as network security, information security and workstation security, is facing a paradigm shift with the ever-increasing gain in popularity of mobile devices such as smart phones and tablets. As many of the current threats to mobile devices (also known as cell phones or mobile phones) are similar to those that threaten desktop machines connected to the internet, many of the same solutions can be adapted to deal with mobile devices. Nevertheless, mobile devices present their own unique challenges such as a fragmented operating system market (such as Apple Os, Android, Windows mobile, RIM etc.), a proliferation of manufactures building devices on different standards, as well as the more limited processing and data storage capabilities of mobile devices. Security solutions have to be programmed with these limitations in mind.

This migration of computing from desktop devices to smart phones and tablets has lead to the appearance of threats that initially only affected desktop devices. The threat that this paper addresses is the migration of spamming Botnets onto mobile devices. Botnets are now capable of infecting mobile devices and using them to send mobile spam.

The paper is structured as follows: the background section introduces the topics of spam, Botnets and Bayesian spam filtering. The following section introduces a model on combating mobile spam through Botnet detection using Bayesian spam filtering. This is followed by a description of the prototype, the actual implementation of the prototype, a section where experimental results are tabulated. The paper is then concluded.

II. BACKGROUND

A. Spam

Unsolicited bulk mail, otherwise known as spam, is an email (electronic message) sent to a large number of email addresses, where the owners of those addresses have not asked for or consented to receive the email [1]. Spam is used to advertise a service or a product. One of the most well-known examples of spam is an unsolicited email message from an unknown or forged address advertising Viagra [2]. The lack of a universally recognized definition for spam is one of the major obstacles to creating solutions designed to minimize its harmful effects. The definition of spam could range from any unsolicited emails [3], to unsolicited commercial emails sent by any source [4], to unsolicited commercial emails from sources the recipient has never had contact with [5], to simply emails or postings transmitted in bulk quantities [6].

Spam is not limited to e-mail as is usually thought to be the case. Spam also exists in text messaging services (SMS). In the case of an SMS, spam can cost even more that it would when received through email. For example, assume that a user has subscribed to receive a notification via SMS when he/she receives an email. Depending on the particular cellular network, the user might have to pay for every SMS received regardless if it is a spam or a valid email.

B. Botnets

A Botnet is a network that consists of a set of machines that have been taken over by a spammer using Bot software Bot software (or Bots for short) is a kind of malware that is often distributed in the form of a Trojan horse [7]. The challenge for businesses and banks in the near future will be to produce secure mobile applications while ensuring ease of use at the same time [8]. The motivation for installing a Botnet that sends spam on a mobile device and installing one on a desktop differ. In the case of Botnets on

desktops the motivation is to prevent the spammer's identity being revealed, in the case of mobile Botnets that is not the only motive the cost of sending SMS messages is exponentially higher than the cost of sending email messages. Thus another motivation for mobile Botnets is to pass on the cost of sending the message to someone else. Malware writers can also use Bots to send SMSs to premium rate numbers. The following section discusses Bayesian filtering.

III. 2.3 BAYESIAN FILTERING TECHNIQUES

This section gives an overview of Bayesian spam filtering techniques. Firstly an overview of Bayesian filtering is given. This is then followed by an explanation of Bayesian filtering applied to mobile spam.

A. Bayesian filtering

Bayesian spam filtering is a statistical technique that makes use of probabilities to classify email as being spam or non spam [9]. This classification is done by correlating the message contents with spam and non spam messages and then using Bayesian inference [10] to calculate the probability of it being a spam message or not. The process works as follows. Some words will have a particular probability of occurring in spam messages and legitimate messages. The spam filter will not know the probabilities of these words occurring in advance, but is trained to learn the probabilities by use of both black and white lists. Black lists contain message samples that are definitely spam and white lists contain message samples that are definitely legitimate. These black and white lists are built up manually by the user. This is accomplished by adding available spam messages to the black list and available legitimate messages to the white list. The size of the dataset used to train the filter is important as filter performance climbs with increased data to train upon [11]. The size of the dataset used to train the filter in this experiment was limited by the number of legitimate and spam SMSs the authors were able to collect. The data selection process is expanded upon in section 4.2. The following section discusses the application of Bayesian filtering to mobile spam.

B. Bayesian filtering applied to mobile spam

Having a good term representation is one of the most important parts for getting a good classifier [12]. The spam filter should be designed with the SMS message characteristics in mind, namely only 160 characters are allowed on a standard SMS message [12]. The problem with this limitation is that fewer words means less information that can be transferred. Also, due to this constraint, people tend to use acronyms when writing SMSs [12]. SMS messages lack structured fields and their text is rife with abbreviations and idioms [13]. Moreover, the abbreviations used by SMS users are not standard for a language, but they depend on the user communities. Such language variability provides more terms to the word collection. For example, the sender of an SMS could replace the word 'tomorrow' with the abbreviation 'tmw'. This would usually be understandable to the SMS recipient, but there would now be two permutations of the word 'tomorrow' to analyse, hence, expanding the collection of words for which their spam probability need to be calculated.

IV. A MODEL FOR A BOTNET DETECTOR USING BAYESIAN FILTERING

In this section we introduce the model for our spamming Botnet detector and explain the reasons for it being modelled in this manner as well as its advantages and disadvantages. We then round this section off by explaining how the prototype would be used in a real-world situation to combat Botnet spam.

A. Remote analyses vs. Analysis on the device

There are two ways in which one could analyse a mobile device user's SMS data. The first possibility would be to analyse the SMSs sent on the ISP's servers and use this to build a profile of the user's SMS sending behaviour. There are, however, several drawbacks to this approach. Firstly, there are privacy implications of analysing SMS data on an ISPs server, but even if these concerns were addressed, the classification algorithm would have to determine whether or not a message is valid or invalid without the user's input. Thus, it would not be able to learn based on user feedback.

The second solution is to implement the detection algorithm on the device, thus, taking care of the privacy implications as well as allowing for user feedback in the learning process. The major disadvantage of this approach is that the prototype needs to be installed on a mobile device which has limited processing power and storage space, especially when compared to an ISP's server. Thus, the prototype had to be programmed to use minimum storage and be optimised to make use of as little memory as possible.

B. Flow Diagram for Botnet Detector

The detection of SMS spam is a typical classification problem where patterns, also known as signatures, need to be classified as legitimate or not. Thus, it is a binary classification problem in which the data set,

consisting of messages, are classified as either spam or non spam. Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate and non-legitimate (spam) messages. From this data the filter should be able to deduce whether messages are spam or not. Figure 2 visualizes how the Botnet detector is designed to work. The mobile user enters a text message and sends it to a recipient. This message is intercepted and analysed by the BBD, which then determines whether the message is valid. If the BBD can determine that the message is valid, the message is sent onwards. Else the network provider will be alerted by the BBD so that the service provider can investigate the malware that is sending these spam messages and remove it from the mobile device.

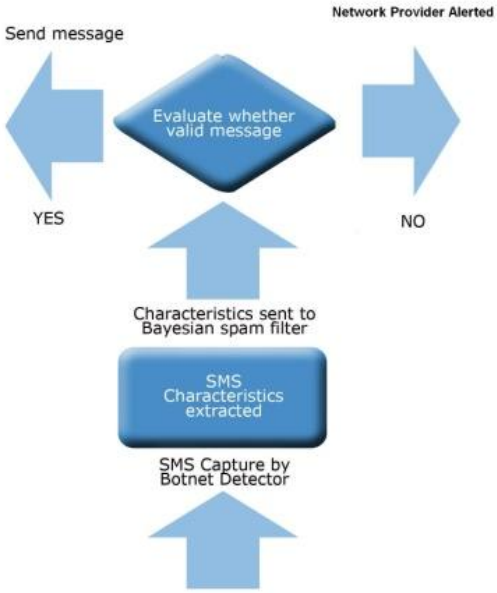

Figure 2: Model for the Botnet detector

The premise behind the BBD implementation is that the authors believe that a spamming Botnet that has installed itself on a user's mobile device, and is sending out spam SMSs without the knowledge of the mobile devices user, can be detected by the BBD. Thus, if these spam messages are blocked, the BBD would have succeeded in preventing the sending of spam as well as saving the mobile device owner the cost of the spam SMSs being sent. Additionally, the BBD would alert the mobile device user to the presence of Botnets on their device so that the malware, that has installed itself on their device, can be removed.

V. IMPLEMENTATION OF SPAMMING BOTNET DETECTOR – A PROTOTYPE

This section describes how the prototype was implemented. The authors discuss how the algorithm used to train the BBD was selected, how the dataset used to train the BBD module was chosen, and how the Bayesian spam filtering was implemented.

A. Algorithm selection

As discussed in section 3, the detection of SMS spam is a typical classification problem where patterns need to be classified as legitimate or not. Thus, it is a simple binary classification problem in which the data set (messages) are classified as either spam or non spam. As stated in section 3, Bayesian filtering is ideal for this sort of classification problem as we can train the spam filter on legitimate (white list) and non legitimate (black list) messages and from this data the filter should be able to deduce whether messages are spam or not.

The algorithm selected is an implementation of Paul Graham's original Naïve Bayesian Spam filtering algorithm [11]. The statistical approach to detecting spam used by the BBD does not attempt to identify the individual properties of spam, such as the words "click on the link". Rather, we leave it up to the algorithm to determine the probability of a message being spam or not based on the datasets used to train the filter on. Unlike feature-recognising spam filters like SpamAssassin [11], which assign a score to messages, the BBD, which uses Bayesian filtering, assigns a probability to a message as an indication of it being spam or not. Because it is measuring probabilities, the BBD considers all the evidence in the SMS message, both good and bad. Words that occur disproportionately rarely in spam, like "later" or "morning", contribute as much to decreasing the probability as possible spam words like "click" and "opt-in" do to increasing it. Therefore, an otherwise innocent SMS message that happens to include the word "Viagra" is not going to be misidentified as spam.

B. Data selection

The data that was used to train the BBD implementation was selected by using SMS messages collected by the SMS Spam Collection project [15] and from the Grumbletext website [14]. Grumbletext is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the spam message received.. In total 4815 of these valid SMSs were used to create a white list of legitimate SMS messages. The second

set, i.e. the black list of invalid SMSs, totalled 746 (including blacklisted URLs).

C. Prototype setup

When the BBD initialises, it reads through the white and black lists and builds a list of words. It then iterates through every word in the list and looks up its individual spam probability. Figure 3 shows an example set of words and their spam probabilities. The closer the probability value is to 1, the greater the probability of it being spam. The closer the probability value is to 0, the greater the probability of it being a legitimate SMS message.

```
.0218,would
.4691,Would
.3293,wow
.9861,www
.9998,x150p
.9998,XCHAT
.5956,Xmas
.9998,XMAS
.2866,xx
.5956,XX
.3616,xxx
.4010,XXX
.8154,xxxxxxx
.0427,ya
.0256,Yeah
.2164,year
.2903,years
.9999,YES
.0686,yesterday
.0602,yet
.1971,yo
.0655,Yo
.1402,you
.3673,You
```

Figure 3: List of random words and their spam probabilities

After the BBD was built, we could start testing it by feeding it valid and invalid SMS messages. Using Paul Graham's original Naïve Bayesian Spam filtering algorithm [11] we process the SMS message body by iterating through every word in the SMS body, and looking up its individual spam probability. We then sort this list of spam probabilities in descending order based on how far our probability is from 50%.

Once our list is compiled and sorted we then combine the most "interesting" probabilities together into to give us a single probability which is used to determine whether the whole message is spam or not. In this case "interesting" is defined as how far our score is from 50%, i.e. how close the score is to indicating a word is spam or valid. Equation 1 gives us the equation used to combine the individual probabilities. In section 5 this is explained further with a real example.

$$\frac{abc\ldots n}{abc\ldots n + (1-a)(1-b)(1-c)\ldots(1-n)}$$

Equation 1: Formula to combine probabilities

D. Running the BBD Prototype

To test the BBD the authors extracted 86 spam SMSs to test the BBD from the Grumbletext website [14]. The authors then extracted another 86 valid messages from one of the author's own phones to test for false positives. False positives refer to legitimate SMS messages that are mistakenly identified as spam. As discussed in the previous section, the spam filter works by first computing the spam probabilities of each individual word in a message and then combines the most "interesting" probabilities together to give us a single probability for the entire message. To determine the best number of "interesting" probabilities to combine, we tried various combinations. The combinations we experimented with were 15, 10, 5, and 3. These combinations are explained further in section 5

VI. Tabulation of Results and Discussion

The results of the experiment are tabulated as follows and show the accuracy in detecting spam SMSs in Table 1.

Number of Combinations	**Correctly identified as spam**	**Incorrectly not identified as spam**
15	(74%)	(26%)
10	(74%)	(26%)
5	(82%)	(18%)
3	(87%)	(13%)

Table 1: Results of running BBD on spam messages.

As can be seen from the results in Table 1, reducing the number of probabilities we combine allows us to better identify a message as spam (true positive). This is intuitively what we would be expecting since the number of words in an SMS message usually ranges between 15 and 25 words, because SMS messages are limited to 160 characters (these numbers were obtained by counting the average number of words in a random selection of SMS messages sent by one of the authors over the last six months). Most email spam filters would combine the top 15 - 20 most

interesting probabilities [11]. As SMS messages are generally much shorter than emails, it would make sense to combine a smaller set of probabilities to evaluate whether or not a message is spam. Figure 4 shows us the BBD's performance in accurately identifying spam based on the number of combinations.

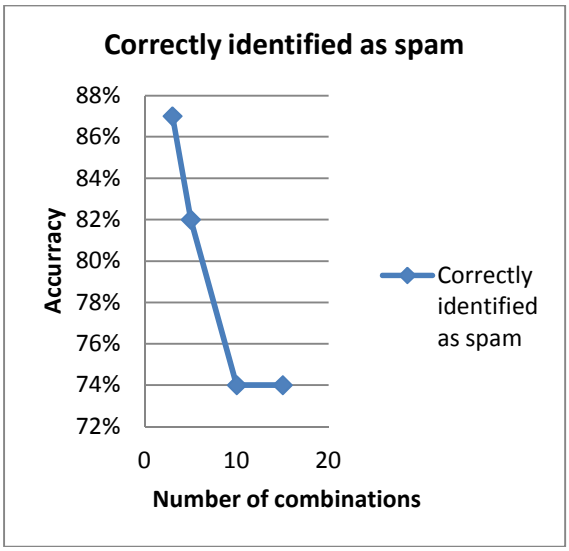

Figure 4: Visualises the effect reducing the number of combinations has on the BBD's performance

It should, however, be noted that most of the spam messages that were not correctly identified as spam were loaded with words that are very similar to SMSs that one of the authors would normally send. Figure 5 is an example of a message that was not correctly identified by the BBD as being spam.

```
your secret admirer is very caring about his friends and holds them dear to his heart. He will do anything for a close friend
```

Figure 5: An example spam message not picked up by the BBD

To expand this point further, let us consider the spam probabilities of the words that occur in this SMS in closer detail. Figure 6 is an example spam SMS message with each word's spam probability.

```
your:0.422130102351602
secret:0.702064402663205
admirer:0.9998
is:0.245045552028142
very:0.0409412132430149
caring:0.011
about:0.0631279200737555
his:0.0275424076854587
friends:0.158827296392993
and:0.187233877650676
holds:0.187233877650676
them:0.0355117066206902
dear:0.011
to:0.368514614315756
his:0.0275424076854587
heart:0.0756322233148982
He:0.011
will:0.162777015101
do:0.0605925536861735
anything:0.011
for:0.342251411689513
close:0.184724884863709
friend:0.304517249692058
```

Figure 6: An example spam SMS message with each word's spam probability

As was discussed in section 4.4, the spam probabilities of a selection of "interesting" words are combined to give us a single probability. The top three most "interesting" words in this example are admirer: 0.9998, caring: 0.011 and He: 0.011. These three probabilities combined as per Equation 1 are given in equation 2.

$$\frac{(0.9998)(0.011)(0.011)}{(0.9998)(0.011)(0.011)+(1-0.9998)(1-0.011)(1-0.011)}$$

Equation 2: Formula to combine probabilities

The result of equation 2 is 0.0068311747673944. A number close to one indicates that the message is spam and a number close to zero indicates that the message is valid. Thus, the message is incorrectly identified as non spam. This example shows us that it is possible for some messages to get through the spam filter, but as shown by the example it is not obvious even to a human filter that the message is spam.

We will now look at how well the BBD works in identifying legitimate messages by measuring the number of false positives, i.e. the number of legitimate messages, incorrectly identified as spam. Table 2 shows the results of the experiment when testing for legitimate messages.

Number of Combinations	**Correctly identified as legitimate**	**Incorrectly identified as spam**
15	(100%)	(0%)
10	(100%)	(0%)

5	(100%)	(0%)
3	(100%)	(0%)

Table 2: BBD results after testing on legitimate messages.

As can be seen from the results in Table 2, the number of false positives is zero.

VII. CONCLUSION AND FURTHER WORK

Spamming Botnets have the potential to become more common place. With the increasing processing power of mobile phones, they are becoming more attractive for exploitation by malware. The aim of this research is to provide a tool that is not only capable of identifying spam SMSs being sent from a user's mobile device, but also to allow the spam filter to learn new spam features as spammers continuously change the spam features to confuse spam filters.

The BBD was capable of correctly identifying 87% of all Spam messages. The authors believe that over time, with more spam messages added to the black list, the accuracy of this implementation would improve. The authors will attempt to apply these ideas on an Instant Messaging platform as well as on SMS messaging in future research.

This BBD, as stated in the paper, could be implemented either on a mobile device or on a network provider's server. The advantage of installing it on a mobile device is that the user can be prompted to confirm whether a particular message is indeed spam or not in the case that the neural network is unsure. The disadvantage of installing the BBD on a user's mobile device is the degradation of performance that could be experienced when the BBD calculates its spam probability. The advantage of situating the BBD on a network service provider's server is that processing can be done by dedicated servers. The disadvantage, of course, is that the user cannot be prompted for feedback and confirmation, which might result in certain messages being incorrectly classified.

VIII. ACKNOWLEDGMENTS

This work is based on research supported by the National Research Foundation of South Africa (NRF) as part of a SA/Germany Research cooperation programme. Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore the NRF does not accept any liability in regard thereto

IX. REFERENCES

[1] Internet Service Providers' Association, 2008. 'What is Spam?' Available: http://www.ispa.org.za/spam/whatisspam.shtml. [April 2009]

[2] Spam-site "Samples of Spam" http://www.spam-site.com/spam-sample.shtml [September 2011]

[3] B.G. Kutais, "Spam and Internet Privacy", 'Journal of High Technology Law Suffolk University Law School'

[4] Consumer fraud reporting, "Spam Emails and Spamming", http://www.consumerfraudreporting.org/spam.php [September 2011]

[5] Federal Communication Commissio, "Spam: Unwanted Text Messages and Email", http://www.fcc.gov/guides/spam-unwanted-text-messages-and-email [September 2011]

[6] Spamhaus "The Definition of Spam", http://www.spamhaus.org/definition.html [September 2011]

[7] Seach Security.com "botnet (zombie army)". http://searchsecurity.techtarget.com/definition/botnet [September 2011]

[8] Emerging Cyber Threats Report for 2009, Georgia Tech Information Security Center, October 15, 2008

[9] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz (1998). "A Bayesian approach to filtering junk e-mail". AAAI'98 Workshop on Learning for Text Categorization.

[10] Box, G.E.P. and Tiao, G.C. (1973) Bayesian Inference in Statistical Analysis, Wiley, ISBN 0-471-57428-7

[11] Paul Graham (2003), Better Bayesian filtering

[12] José María Gómez Hidalgo , Guillermo Cajigas Bringas , Enrique Puertas Sánz , Francisco Carrero García, Content based SMS spam filtering, Proceedings of the 2006 ACM symposium on Document engineering, October 10-13, 2006, Amsterdam, The Netherlands [doi>10.1145/1166160.1166191]

[13] Gordon V. Cormack , José María Gómez Hidalgo , Enrique Puertas Sánz, Feature engineering for mobile (SMS) spam filtering, Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, July 23-27, 2007, Amsterdam, The Netherlands [doi>10.1145/1277741.1277951]

[14] Grumbletext, Available: http://www.grumbletext.co.uk/vt.php?t=333&subj=complaints++SMS+Spam+%28General%29+complaint, accessed 24/02/2012

[15] SMS Spam Collection, Available: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/, accessed09/05/2