Content Security and Privacy Preservation in Social Networks through Text Mining

José María Gómez Hidalgo[1], José Miguel Martín Abreu[1],
Pablo García Bringas[2], Igor Santos Grueiro[2]

[1] Optenet, José Echegaray 8, 28230 Las Rozas, Madrid, Spain
{jgomez, jabreu}@optenet.com
[2] S3Lab, Deustotech, Universidad de Deusto,
Avda. Universidades 24, 48007, Bilbao, Bizkaia, SPAIN
{pablo.garcia.bringas, isantos}@deusto.es

Abstract. Due to their huge popularity, Social Networks are increasingly being used as malware, spam and phishing propagation applications. Moreover, Social Networks are being widely recognized as a source of private (either corporate or personal) information leaks. Within the project Segur@, Optenet has developed a number of prototypes that deal with these problems, based on several techniques that share text mining as the underlying approach. These prototypes include a malware detection system based on Information Retrieval techniques, a compression-based spam filter, and a Data Leak Prevention system that makes use of Named Entity Recognition techniques.

Keywords: Text Mining, Information Retrieval, Named Entity Recognition, Malware, Spam, Data Leak Prevention

1 Introduction

As Social Networks (SNs) gain more and more popularity, they are becoming a major source and propagation vector for malware, spam, phishing and private information leaks. It is imperative to control these threats in order to make SNs as useful as they promise to be. The project Segur@ has helped Optenet to frame a number of research developments addressing these problems, and sharing Text Mining as the base of them. These developments, along with the prototypes developed, are described in the next sections.

2 Malware Detection using Information Retrieval with Opcodes

Malware (trojans, spyware, viruses, etc.) is plaguing SNs, mostly in the form of obfuscated variants of an original malicious software that adapts to the current user machine specifics. In order to detect these new variants and even new malware families (i.e., completely new malicious software), already known software (both malicious and legitimate) can be represented in terms of operational codes (assembler code instructions), and new software similarity to these known executables can be computed. We have designed an approach in which:

1. Software is represented as sequences of operational codes (opcodes) of several lengths, after selecting the most relevant ones according to its Mutual Information with respect to being malware or legitimate software. Opcode n-grams are terms in a Vector Space Model, weighted using a weighted term frequency formula.
2. New software instances similarity with respect to malware or legitimate software is computed using the cosine similarity between the vectors that represent all software instances. A new instance is considered malware or legitimate depending of its similarity to both kinds of instances. This way, variants of currently known malware are detected.
3. A Machine Learning approach is used to derive a classifier able to detect new malware instances that share properties with existing code samples. Several Machine Learning approaches have been tested, including Support Vector Machines (SVMs), Decision Trees, or k Nearest Neighbour classifiers.

Overall, a Text Classification approach is used, but on assembler code instead of text. The experiments carried out for these approaches are very encouraging, as the results obtained on a test collection of over 26,000 samples of malware and legitimate software show that:

1. It is quite easy to find a discriminative threshold between malware and legitimate software in the detection of variants of known malware instances when using opcode bigrams, leading to high detection rates and no false positives.
2. For instance, SVMs and Decision Trees (Random Forests) over one and bigrams lead to detection rates over 95% with false positives below 5% by cross-validation on the test collection.

A prototype has been developed in order to demonstrate the analysis process on a Microsoft Windows Systems. The prototype allows loading a software instance, disassembling it, and comparing it to the current database of software samples.

3 Compression-based Spam Detection

SNs are being used as a prominent method for disseminating spam and phishing attacks, what represents a major threat for SNs users. A wide range of learning based systems and approaches for spam detection (Bayesian filters) have been developed in recent years. Despite of their theoretical effectiveness, most of them are very sensitive to the tokenization process, in which messages are represented as sequences of character strings intended to capture either message semantics or spam properties.

State of the art on Bayesian filters as demonstrated in the TREC Spam Filtering Tracks [1] shows that compression based learning, often applied to Text Categorization (see e.g. [3]), does not face the same problem. The reason for this is that messages are not explicitly represented in terms of strings resembling works, but in terms of bit sequences hardly hacked by spammers in order to overcome the filter.

We have developed a compression based spam filter that includes a number of text compression approaches [1], namely Dynamic Markov Chains, Prediction by Partial Matching, and Lempel-Ziv variants as GZip. The compression filter is able to classify email messages (as those received through SNs with updates and status summaries) as spam or legitimate, and it has been implemented resembling Spamassassin command line operations, in order to simplify its integration with Mail User Agents like Thunderbird or KMail over Linux.

4 Privacy Preservation through Named Entity Recognition

Perhaps the most dangerous current security threat for SN users is private information leaks. From phishing attacks to unnoticed users mistakes, leaks are getting more and more common, and Data Leak Prevention (DLP) tools must be made more and more effective in order to preserve user and corporate privacy.

Most DLP tools are very effective when protecting already known private information. However, a great amount of private information is not recognized until it has been disclosed to unknown users or competition enterprises. In order to detect unknown private information leaks, we have designed a system based on:

1. Text Classification techniques that are used to process any text disclosed to a SN in search for Named Entities like person, organization or product names. Language Independent techniques (e.g., n-gram features) are used to detect unknown patterns (entities), using the package Freeling [2] for testing our approach on Spanish and English text instances.
2. A user-learning loop similar to that of personal firewalls. The user is alerted when a new entity is detected, allowing them to classify as private or public, and to block or allow text including the detected entity to be posted to the net. Subsequent occurrences of the entity are blocked or allowed by using a white/black list. In consequence, the software gradually adapts its performance to the user/organization behaviour.

The Named Entity Recognition module of our prototype has been tested on a sample of entities and text postings collected over a year on the micro-blogging Twitter SN, covering a range of popular brand and product searches according to Google Insights statistics. Results of our experiments show accuracy over 90% in English and Spanish, and very low false negative rates. Moreover, most false negative text instances (which correspond to undetected entities which may lead to private information leaks) are detected in other instances, and in consequence, the user is alerted regarding them anyway.

The prototype configuration screen is presented in Fig. 1, showing the whole range of privacy enabling data fields (from credit card numbers to passwords, etc.). Named Entities detected are stored as user-defined fields (*Other Keys* section). The prototype is available for Microsoft Windows.

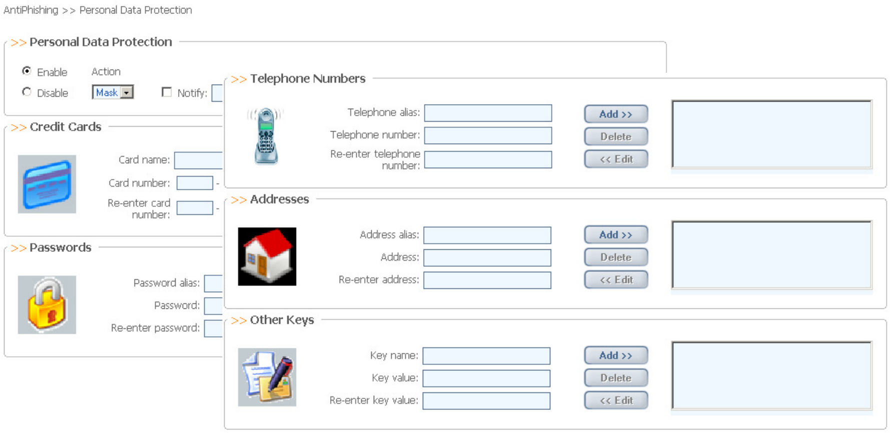

Fig. 1. Data fields used in privacy protection features. Named Entities are stored in the Other Keys section, as the users are getting alerted and they decide to supervise them.

5 Future Work

Current prototypes future work plans include improving both the text analysis techniques involved, and getting the user interface getting integrated into the Optenet Security Suite for PC, enabling current and future Optenet users to benefit from these advances.

Acknowledgments. The prototypes described in this paper have been developed within the Project Segur@ (CENIT-2007 2004).

References

1. Cormack, G.: TREC 2007 Spam Track Overview. In Voorhees E.M., Buckland, L.P. (eds): NIST Special Publication 500-274: The Sixteenth Text REtrieval Conference Proceedings (TREC 2007).
2. Freeling Home Page, http://www.lsi.upc.edu/~nlp/freeling/.
3. Teahan, W.J., Harper, D. J.: Using compression based language models for text categorization. In Callan, J., Croft, B., Lafferty, J., (eds): Workshop on Language Modelling and Information Retrieval, pages 83--88, Carnegie Mellon University, 2001.