Soft Computing Techniques Applied to Corporate and Personal Security

Paloma de las Cuevas
Dept. of Computer Architecture and Computer Technology
University of Granada, Spain
paloma@geneura.ugr.es

Juan Julián Merelo
Dept. of Computer Architecture and Computer Technology
University of Granada, Spain
jmerelo@geneura.ugr.es

Pablo García-Sánchez
Dept. of Computer Architecture and Computer Technology
University of Granada, Spain
pablogarcia@ugr.es

ABSTRACT

Inside a "Bring Your Own Device" environment, the employees can freely use their devices. This allows them mix their personal and work life, but at the same time, if the users are not aware of a risky situation, or that situation is not covered by a company security policy or rule, this environment can become very insecure. The aim of this paper is defining a novel system architecture able to self-adapt itself, in the sense that it will learn from past, non secure situations, and therefore will be able to determine whether a new situation is risky or not.

This Paper proposes the use of a variety of techniques, from Data Mining of big amounts of recorded data to Evolutionary Algorithms for refining a set of existing policies, maybe creating new ones. A preliminary method that automatically extracts rules to avoid or deny URL connections helps to demonstrate that, by performing a good preprocessing of the data, useful conclusions can be extracted from new - unknown - situations. Therefore, it is possible to successfully extend a set of rules, usually laid out by the company, for covering new, and potentially dangerous, situations.

Categories and Subject Descriptors

H.2 [**Information Systems Applications**]: Database Management; H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Data Mining; Corporate Security Policies; Evolutionary Algorithms; Machine Learning; Classification

1. INTRODUCTION

The evolution from traditional mobile phones to the so-called smartphones has changed the way people use their devices. In addition, security threats have evolved too [11], so that new security measures have to be adopted every time a new threat appears. More specifically, smartphones have contributed to the creation of a *Bring Your Own Device* (BYOD) scenario in which people use their own devices at work. Despite of all the advantages that this environment might have, it is clear that this kind of situation creates new security challenges for the Chief Security Officer (CSO) of a company [21]. This is because they want a fast response to any user action that might cause harm (in terms of money loss because of a security incident) to the company, but without monitoring the users in a way that is against privacy. The task of the CSO and the security department of a company, is establishing a list of security measures to cope with all the security incidents which might happen in every environment, so they build what is called a set of *Corporate Security Policies* (CSPs). These are a set of security rules aiming at protecting company assets by defining permissions for every specific behaviour that could lead to security incidents [14]. But when such companies embrace a constantly changing environment, and allow their employees to use their own devices, the risk of having security incidents grows, even if the employees do not have intention of attacking the company [22, 6]. Then, there is a need of constant renewal of the CSPs, which it might be difficult if new threats appear without knowing them in advance (because they are not included in the security policies).

This paper proposes a system architecture, which should be easily integrable in company servers, and capable of evolving the rules included in a CSP by learning from past user behaviours which caused security incidents. In order to achieve this, different techniques have to be applied. First, we assume that a company stores the security incidents that have been produced, along with the context in which they were produced. Context was defined by Abowd et al. [1] as "any information that can be used to characterize the situation of an entity". Then, and given that this means to analyse great quantities of data, Data Mining (DM) techniques can help to extract useful information from it [9], and also from what can be considered as *good behaviour* (this means, actions that were permitted by the security rules). This process would allow to build a classifier and to further classify new situations. With the extracted conclusions from the performed DM analysis, new rules can be automatically inferred. Then, as rules can be seen as a tree whose branches are the conditions, and whose leaves are the rule decisions, an Evolutionary Algorithm (EA) can be applied to optimise its structure.

The paper is structured is as follows. A brief state of the

GECCO '15, July 11 - 15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739482.2768477

art in company security systems and the use of DM and EAs on them is given in the next section. Then, Section 3 explains the overview of the architecture of the proposed system. This system will allow the automatic creation of new security rules, as well as optimisation of the existing ones, for a faster response to new - and potentially dangerous - events. Previous results obtained over a particular type of data - URL connections - in order to evolve black and white URL lists are presented in Section 4. Finally, conclusions and future work are shown in Section 5.

2. STATE OF THE ART

Many tools for companies, as well as for devices, which have adopted the BYOD have been released in the past four years. This way, and more focused on the enterprise, tools such as IBM's Hosted Mobile Device Security Management or Sophos Mobile Control offer the CSO ways to control the devices which enter in the system, requiring users to employ strong passwords, for instance, and also to protect the employees data by means of data encryption and data protection by having strong and secure passwords . Other tools for managing a BYOD situation, such as the one developed by Good's Technology [24], adds to their features guidelines for the CSOs to develop good CSPs. However, not one of the reviewed tools has the ability of inferring new rules or refining the existing ones.

On the device side, the most powerful solution to protect them in a BYOD situation seems to be to directly use a phone which has been developed with data security in mind such as the BlackPhone [7]. It has its own Android-based operating system, called *PrivatOS*, which includes a privacy-focused application store (called *Silent Store*) that takes care of the problem of applications which ask for certain permissions that can lead, for instance, to personal data leakage [11]. This BlackPhone also allows a remote wiping of the data if the device is lost or stolen. The main disadvantage of this solution is either the enterprise having to make an investment and buy these smartphones to the employees (which, in fact, is against the BYOD philosophy), or to make employees buy them, so they cannot use the device they already have. On the contrary, the system proposed in this work is designed as device-independent.

Finally, and considered as an extension of Android devices, two main tools can be found in the market: Samsung KNOX for Samsung devices, and Android Work by Google. Both have most of the same advantages as the blackphone, with the addition of an extension for CSOs. This means that Samsung, as well as Google, provide security tools both at device and server side. More precisely, Android work follows the way of working that Blackberry phones started with Blackberry Balance, which stands up for having *work* applications and *personal* applications. This is called a "dual-persona" smartphone [4]. However, with regard to CSPs, neither of these tools specify "self-adaptation' ' as a feature. They offer policy management, but still they do not analyse the system information for security rules evolution purposes.

With respect to the application of Data Mining to extract information from big amounts of data, this has been done since the nineties [3, 10]. More specifically, DM has been widely used for security purposes, as it can be applied in computer forensics. O. de Vel studied the application of DM techniques to identify authors of malicious e-mails in [9], and for performing "offender profiling" in relation to computer security attacks in [2]. Yet, the system this paper proposed is focused in doing this kind of analysis but then to look for similarities with the new incoming events, so that a decision can be made in case they are dangerous. Classification methods are also applied in the security field. For instance, Blanzieri and Bryl [5] present a review on a variety of spam filtering methods, and compare them, reaching the conclusion of that they are successful in general, but yet insufficient. This is why implementing a self-adaptive system such as the one this paper proposes can be good for other security applications and not only spam classification.

As for the works related with the users' information and behaviour, and the management (and adaptation) of the set of Corporate Security Policies, many can be found in literature. For instance, P.G. Kelley et al. [15] presented a method named *user-controllable policy learning* in which the user gives feedback to the system every time it applies a security policy. Then, these policies can be refined according to that feedback to be more accurate with respect to what the users need. This approach could be useful for adding information to the system, and therefore perform a deeper analysis to extract more accurate conclusions, and finally create better rules. Then, taking into account how much information can be gathered from social networks, Danezis in [8] defined a system able to infer privacy-related restrictions, enhancing user's privacy, by applying Machine Learning techniques on a social network environment. Again, this is another interesting approach. However, this paper focuses on CSPs, related to companies, more than on personal life of individuals.

In the same line, Lim et al. proposed a system [18, 17] which evolves a set of computer security policies by means of Genetic Programming, gathering knowledge from the user's feedback like in [15]. Furthermore, Suarez-Tangil et al. [23] take the same approach as Lim et al., but also including event correlation in. These two latter author's works are interesting for this paper, though they are not focused on company CSPs.

Next section describes the methodology which the system proposed in this work will follow. The development of this system is supported by previous experiment that are explained in Section 4.

3. METHODOLOGY

The proposed system is intended to be placed inside the server of the company which wants to add a rule-refinement feature to its security system. Furthermore, this self-adaptive system can be seen as a feature extension of the tools described in the previous section. Figure 1 shows an overview of the architecture components of the proposed system.

In order to understand the flow of information, it must be noted that the *database* represents only the part inside the company server where the needed data is stored. This also contributes to preserve privacy, for the system would only have rights to access some piece of information. The following subsections describe the two main components of this system: the *data mining analyser*, and the *rule treatment* component, which will use Evolutionary Algorithms for creating and evolving security rules.

3.1 Data mining analyser

This component will be in charge of taking the desired 'raw' data from the database, and processing it to remove

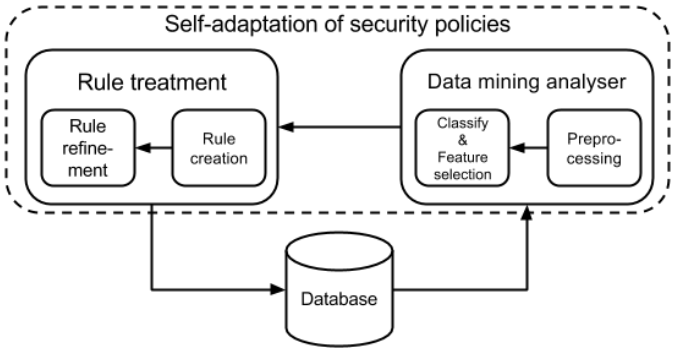

Figure 1: Architecture overview of the proposed system, whith its inner components.

errors or non-valid values, in order to obtain a dataset to be used in the rule treatment process. For instance, duplicated data or unknown values are considered as data that should be removed. Considered data corresponds to events (and their related information/context) produced by users' interactions with the system. Then, the preprocessing component will be devoted to 'prepare' this data for the application of further techniques such as pattern mining [13]. Pattern mining allows the identification of non-frequent or anomalous patterns, since these are suspicious, and thus, could be of interest to be checked by the Chief Security Officer.

The next subcomponent performs tasks like feature selection [12], which consists of choosing the most important data features/variables in order to reduce the dataset weight. Also, new features can be created by extracting meta information from the existing ones. These two steps are mainly done for improving the performance of the classification stage. Then the subcomponent uses classification algorithms [25], i.e., it trains models (classifiers) able to associate every pattern in the dataset to a class. This way, the built classifier can assign a class to further incoming patterns. In this case, the class will be the "decision" taken, which means that if the incoming user action is too similar to past dangerous patterns, it will be rejected or denied.

3.2 Rule treatment

This component will be focused in creating new rules and will also work with the existing CSPs. It will globally perform three different tasks over them. First, it will take the set of classification rules from the previous component, and will merge or compare them with the existing ones, suggesting a first set of new (unrefined) rules. Then, it will analyse the existing rules in order to remove, from the created set in the first step, those which might be redundant. This is done for maintaining correctness and coherence in the system. Finally, taking advantage of the decision tree structure nature of the rules, the system will consider using Genetic Programming [16], as the kind of EA used for optimising tree-based structures. Furthermore, the system can determine if to make a certain rule more specific, or general, would be more efficient, and would cover more incidents. The final set of rules will be presented to the CSO of the company, who will accept or reject them. This acceptance or rejection of rule process is itself a 'feedback' from wich the system can learn.

4. PRELIMINARY RESULTS

The decision of implementing this system is preceded by the results obtained in [19]. In that work, the type of events that the authors worked with were related to URL requests, that is, users making URL requests to the company proxy server. Then, instead of evolving Corporate Security Policies by themseves, the authors demonstrate that it is possible to extend the performance of what are called as black/white lists (non permitted/permitted URLs). This extension, however, is not aimed for including new URL strings, but to obtain a set of rules wich classify a certain URL request by other parameters such as the type of content of the webpage, or the size of the content.

Then, in this work a dataset of 100000 patterns about employees' URL sessions information is analysed. Considering a set of URL access permissions (initial black/white lists), they compose a labelled dataset over which they test several classification methods, applying different preprocessing processes over the data each time. The results show that classification accuracies range from 95% to 97%, but more importantly, that the rules obtained from the output of the trained classifiers are useful for classifying future URL requests, even if the specific URL string is not included in a black or a white list.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a proposed system architecture for evolving a set of security policies and rules, either creating new rules or refining the existing ones, for a company to be able to better secure its assets when adapting to a *Bring Your Own Device* environment. This kind of environment can be dangerous because the users use their own devices for work, either inside or outside the workplace, and they are not always aware of being involved in risky situations. A risky situation is any situation in which the low level of security might cause a security incident, which itself is translated in money loss for the company.

Then, as the Corporate Security Policies of a company cannot cover situations that never happened before, the system proposed in this work is able to extract knowledge from past situations, either dangerous or not, and can optimise the set of CSPs by creating new rules, refine them, or generalise them.

Also, previous results obtained over real data of a particular type of event, which is a user making a URL request,

support the idea of developing a system like the one proposed.

As future work, the system will be implemented and tested in a real system, of a real company, as part of the MUSES European project prototype trials [20].

6. ACKNOWLEDGEMENTS

This paper has been funded in part by European project MUSES (FP7-318508), along with the Spanish National project TIN2014-56494-C4-3-P (EphemeCH).

7. REFERENCES

[1] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[2] Tamas Abraham and Olivier de Vel. Investigative profiling with computer forensic log data and association rules. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 11–18. IEEE, 2002.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

[4] Ron Amadeo. A review of Android for Work: Dual-persona support comes to Android, 2015.

[5] Enrico Blanzieri and Anton Bryl. A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1):63–92, 2008.

[6] GF Breivik. Abstract misuse patterns - a new approach to security requirements. *Master Thesis. Dept of Information Science. Bergen, University of Bergen, N-5020 NORWAY*, 2002.

[7] Silent Circle. Blackphone website, 2014.

[8] George Danezis. Inferring privacy policies for social networking services. In *Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence*, AISec '09, pages 5–10, New York, NY, USA, 2009. ACM.

[9] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Record*, 30(4):55–64, 2001.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96*, pages 226–231, 1996.

[11] A. Gangula, S. Ansari, and M. Gondhalekar. Survey on mobile computing security. In *Modelling Symposium (EMS), 2013 European*, pages 536–542, Nov 2013.

[12] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[13] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[14] Merike Kaeo. *Designing network security*. Cisco Press, 2003.

[15] Patrick Gage Kelley, Paul Hankes Drielsma, Norman Sadeh, and Lorrie Faith Cranor. User-controllable learning of security and privacy policies. In *Proceedings of the 1st ACM Workshop on Workshop on AISec*, AISec '08, pages 11–18, New York, NY, USA, 2008. ACM.

[16] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[17] Yow Tzu Lim, Pau Chen Cheng, John Andrew Clark, and Pankaj Rohatgi. Policy evolution with genetic programming: A comparison of three approaches. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1792–1800. IEEE, 2008.

[18] Yow Tzu Lim, Pau Chen Cheng, Pankaj Rohatgi, and John Andrew Clark. MLS security policy evolution with Genetic Programming. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1571–1578. ACM, 2008.

[19] A.M. Mora, P. De las Cuevas, and J.J. Merelo. Going a step beyond the black and white lists for url accesses in the enterprise by means of categorical classifiers. In Agostinho Rosa, Juan Julián Merelo, and Joaquim Filipe, editors, *ECTA 2014 - Proceedings of the International Conference on Evolutionary Computation Theory and Applications*, pages 125–134, 2014.

[20] A.M. Mora, P. De las Cuevas, J.J Merelo, S. Zamarripa, M. Juan, A.I. Esparcia-Alcázar, M. Burvall, H. Arfwedson, and Z. Hodaie. MUSES: A corporate user-centric system which applies computational intelligence methods. In Dongwan Shin et al., editor, *29th Symposium On Applied Computing*, pages 1719–1723, 2014.

[21] R. Oppliger. Security and privacy in an online world. *IEEE Computer*, 44(9):21–22, September 2011.

[22] Jeffrey M Stanton, Kathryn R Stam, Paul Mastrangelo, and Jeffrey Jolton. Analysis of end user security behaviors. *Computers & Security*, 24(2):124–133, 2005.

[23] Guillermo Suarez-Tangil, Esther Palomar, José María de Fuentes, J Blasco, and Arturo Ribagorda. Automatic rule generation based on genetic programming for event correlation. In *Computational Intelligence in Security for Information Systems*, pages 127–134. Springer, 2009.

[24] Good's Technology. Good's technology byod solution, 2012.

[25] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.