Applied Soft Computing 11 (2011) 4321–4331

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Analysis of vulnerability assessment results based on CAOS

G. Corral*, A. Garcia-Piquer, A. Orriols-Puig, A. Fornells, E. Golobardes

Grup de Recerca en Sistemes Intel·ligents, La Salle, Universitat Ramon Llull, Quatre Camins 2, Barcelona, Spain

ARTICLE INFO

Article history:
Received 27 April 2010
Received in revised form
14 September 2010
Accepted 15 September 2010
Available online 13 November 2010

Keywords:
Information system security
Multiobjective optimization
Evolutionary algorithm
Unsupervised learning
Clustering
Network security
AI applications

ABSTRACT

Information system security must battle regularly with new threats that jeopardize the protection of those systems. Security tests have to be run periodically not only to identify vulnerabilities but also to control information systems, network devices, services and communications. Vulnerability assessments gather large amounts of data to be further analyzed by security experts, who recently have started using data analysis techniques to extract useful knowledge from these data. With the aim of assisting this process, this work presents CAOS, an evolutionary multiobjective approach to be used to cluster information of security tests. The process enables the clustering of the tested devices with similar vulnerabilities to detect hidden patterns, rogue or risky devices. Two different types of metrics have been selected to guide the discovery process in order to get the best clustering solution: general-purpose and specific-domain objectives. The results of both approaches are compared with the state-of-the-art single-objective clustering techniques to corroborate the benefits of the clustering results to security analysts.

1. Introduction

The increase of the dependency of organizations on information and communication technologies, together with the need of securing companies systems in a world were new threats are risks appear daily, has unleased the demand for new and effective security techniques. Therefore, maintaining a proper level of security is a key challenge in current organizations, even when they have the most advanced technology and trained professionals [1]. Consequently periodic security tests—project-oriented risk assessments of information systems and networks through the application of professional analysis on a security scan—are necessary to assure that security does not degrade below an acceptable risk level. One of the most important analysis included in these tests is the vulnerability assessment, i.e., the process followed to identify and quantify vulnerabilities. Both follow a two-step process: test everything possible and generate a concise report.

The cost and the time involved in a security test may limit its depth, so an automation is essential, specially in the analysis of test results. A complete analysis must also coordinate diverse sources of information to support an intelligent response [2]. So security applications demand intelligence to detect malicious data, unauthorized traffic or vulnerabilities [3]. Machine learning can be applied to process the results of vulnerability assessments. The use of unsupervised learning for discovering hidden patterns through the identification of groups of tested devices with similar vulnerabilities has already been presented in *Analia*. This is the analysis module of the framework *Consensus*, a computer-aided system that automates the processes associated to security tests for information systems and networks [4].

Analia helps security analysts in the task of extracting conclusions from data of security tests. This is due to the integration of different clustering approaches and clustering validation techniques. However, two independent steps are needed before extracting conclusions: analysts have to select (1) the clustering approach and (2) the validity index to return the most appropriate solution. Therefore, the best clustering solution depends on the selected validity index, as each index may evaluate different goals. Moreover, the clustering and the index goals may not be aligned. Analysts also ask for a system where configuration parameters not related to their domain, like the clustering technique or the validation index, are provided automatically.

This paper presents a new contribution in the domain of information system security. The drawbacks of the clustering process in *Analia* are solved with the Clustering Algorithm based on multiObjective Strategies (CAOS), an evolutionary multiobjective (EMO) clustering algorithm, to process data of vulnerability assessments. This approach groups tested devices with similar vulnerabilities guided by different goals, as a multiobjective technique allows. So security analysts will obtain the best clustering solution

* Corresponding author.
E-mail addresses: guiomar@salle.url.edu (G. Corral), alvarog@salle.url.edu (A. Garcia-Piquer), aorriols@salle.url.edu (A. Orriols-Puig), afornells@salle.url.edu (A. Fornells), elisabet@salle.url.edu (E. Golobardes).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.09.011

considering different criteria simultaneously. CAOS includes the optimization of the selected criteria in the clustering process itself. Thus analysts will not need to configure any parameter regarding clustering or validity indexes and will be able to focus only on the obtained clustering results, which is their actual concern. Two different configurations for CAOS are studied, depending on the objectives used to evaluate the system: general-purpose and domain-specific objectives. The experimental analysis presented in this paper demonstrates the improvement of clustering results when using the domain-specific objectives with CAOS. Also the process of extracting conclusions from results is simplified, as analysts are now able to extract the best clustering solution and the most adapted to the domain-specific objectives in a single step.

The remainder of this paper is organized as follows. Section 2 describes related work on machine learning in the security domain. Section 3 explains single-objective clustering techniques. Section 4 details our clustering approach CAOS and the different objective functions. Section 5 describes the clustering process in *Analia*. Section 6 summarizes the experimentation and results. Conclusions and further work are given in Section 7.

2. Related work

The increasing frequency of incidents of security breaches in information systems and the ever-increasing reliance of organizations on information technologies involve a constant monitoring of the existing security level for early detection of any negative variation in that control measure. The last IBM Trend and Risk Report provided an account of vulnerability disclosures in the last few years. It stated that the annual vulnerability disclosure rate appears to be fluctuating between 6 and 7 thousand new disclosures each year. The most prevalent primary consequence of vulnerability exploitation continues to be gain access [5]. A study carried out by IDC states that external threats often overshadow the importance of protecting against internal risks [6]. Therefore, periodic security tests are needed to check that security is maintained. *Consensus* is a security testing framework created to aid security managers in these regular tasks [4]. However, these periodical tests generate large volumes of data that have to be processed to give an alarm signal in case new vulnerabilities or security holes are detected.

The huge amount of data produced by security tests has promoted the use of enhanced techniques to recognize malicious behavior patterns or unauthorized changes in information systems or networks [3]. These domains are usually defined by sets of unlabeled examples, and experts aim at extracting novel and useful information about the network behavior that helps them detect vulnerabilities, among others. In this context, clustering appears as an appealing approach that allows grouping network devices with similar security vulnerabilities, thence, identifying potential threats to the network.

Clustering algorithms can be classified into different ways according to many points of view [7,8] such as the relation between the clusters (partitional or hierarchical), how they are structured (center-based, search-based and graph-based, among others), the relation of the class with the cluster (hard clustering and fuzzy clustering, among others), or the criteria used to build the clusters (conventional clustering, ensemble clustering, or multiobjective clustering). If we address the last criteria, conventional clustering [9] is based on optimizing an objective function for assessing the quality of groups of elements. On the other hand, ensemble clustering [10] and multiobjective clustering [11] use a set of objectives to promote the definition of clusters. The main difference between both approaches is the procedure used to build the clusters. Ensemble clustering divides the procedure into two phases: (1) application of many clustering algorithms following different single objectives; and (2) combination of the last results to create the final clusterization. This last step is quite complex to achieve and it is usually inefficient because the objectives are usually contradictory. This is exactly what the other approach does to tackle this drawback: it evolves solutions based on multiple criteria simultaneously.

Several clustering techniques have been applied to the network security domain thus far. For example, *k*-means [12] has been used to group similar alarm records [13] and to detect network intrusions [14]. Self-organizing maps (SOM) [15] have been employed to detect computer attacks [3], network intrusions [16], and anomalous traffic [17]. Despite the success of these applications, all these clustering techniques guide the discovery process with a single criterion. For example, *k*-means minimizes the total within-cluster variance and tends to find spherical clusters [12]. Our case is different, as we are interested in obtaining clusterings that satisfy different criteria. For this purpose, several authors have proposed to run different clustering techniques to obtain different structures, and then, involve the network expert into the process in order to manually select the best structure according to certain predetermined validation methods.

To automatize this process, we propose the use of a multiobjective clustering approach [18], which guides the clustering process with different objectives. There are different techniques for multiobjective optimization such as simulated annealing or ant colony optimization. We base on evolutionary algorithms since they (1) employ a population based-search, evolving a set of optimal tradeoffs among objectives, (2) can be easily adapted to the type of data of our domain, due to the flexible knowledge representation used, and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions. An example of application of a EMO algorithm has been proposed to improve cryptography [1]. Some EMO clustering approaches have been successfully applied to important real-world problems such as intrusion detection [19], formation of cluster-based sensing networks in wireless sensor networks [20], and creation of security profiles [21].

3. Description of single-objective clustering techniques

This section briefly introduces the three single-objective clustering techniques applied herein: *k*-means [22,23], *x*-means [24], and SOM [15]. These techniques are some of the most influential clustering techniques and they have been shown to be able to discover novel clustering structures on a variety of real-world problems.

3.1. k-means

k-means is a simple method that partitions a given training data set into *k* disjoint clusters—where *k* is specified by the user—using the following simple iterative procedure. Initially, it randomly selects *k* instances of the training data set, and each one is considered the *centroid* of one of the clusters. Then, the algorithm iteratively performs the following two steps. First, each example is assigned to the cluster with the closest centroid. The typical similarity measure employed by the system is based on the Euclidean distance. Second, each cluster centroid is reallocated to the center (mean) of all the examples assigned to it. These two steps are repeated until none of the cluster's centroids change.

The *k*-means technique is one of the most influential clustering techniques [25] due to its simplicity and the competent results demonstrated in real-world applications. Nevertheless, the technique also has several drawbacks, some of which are discussed in the following. Firstly, *k*-means requires the user to specify the