Information Sciences 326 (2016) 71–88

Contents lists available at ScienceDirect

Information Sciences

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

XML-AD: Detecting anomalous patterns in XML documents

Eitan Menahem [a,*], Alon Schclar [b], Lior Rokach [c], Yuval Elovici [c]

[a] IBM Cyber Security, Cyber Security Center of Excellence, Ben-Gurion st. Beer-Sheba 8410501, Israel
[b] School of Computer Science, The Academic College of Tel-Aviv-Yafo, Tel Aviv 61083, Israel
[c] Telekom Innovation Laboratories and Information System Engineering Department, Ben-Gurion University of the Negev, Be'er Sheva 84105, Israel

ARTICLE INFO

Article history:
Received 1 December 2013
Revised 23 May 2015
Accepted 4 July 2015
Available online 22 July 2015

Keywords:
XML anomaly Detection
XML security
Machine-learning
Outliers detection
Anomaly-detection

ABSTRACT

Many information systems use XML documents to store data and to interact with other systems. Abnormal documents, which can be the result of either an on-going cyber attack or the actions of a benign user, can potentially harm the interacting systems and are therefore regarded as a threat. In this paper we address the problem of anomaly detection and localization in XML documents using machine learning techniques. We present *XML-AD* – a new XML anomaly detection framework. Within this framework, an automatic method for extraction of feature from XML documents as well as a practical method for transforming XML features into vectors of fixed dimensionality was developed. With these two methods in place, the XML-AD framework makes it possible to utilize general learning algorithms for anomaly detection. The core of the framework consists of a novel multi-univariate anomaly detection algorithm, *ADIFA*. The framework was evaluated using four XML documents datasets which were obtained from real information systems. It achieved over 89% true positive detection rate with less than 0.2% of false positives.

1. Introduction

Today, an increasing number of information systems interact using XML documents. These documents are vulnerable to cyber-attacks or even unintentional mistakes, which can alter their structure and content. Such altered XML documents, especially those that adhere to an XML Schema Definition (XSD), can potentially damage the interacting information systems. Unfortunately, state-of-the-art end-to-end security protocols for XML documents such as XML encryption [15], XML signature [16] and XML-canonicalization [14] provide little protection against such threats. This is mostly because the XML documents are attacked or deformed prior to these protective measures. Since altered XML documents can be rendered as abnormal with respect to the majority of the XML documents in the same domain, we show that anomaly detection can be effectively used for their interception and thus it should be employed before the above-mentioned security protocols.

Extensible Markup Language (XML) [5] is a framework that facilitates the definition of structured markup languages. Data in such languages is described by documents in which all data items are encapsulated by tags. The flexibility of XML makes it especially suitable for: (a) storing data in a structured format; (b) data exchange both in local organizational networks and over the Internet; and (c) data serialization. Although XML documents must conform with an XSD they can vary considerably. Two XML documents that obey the same XSD can be substantially different in the content and structure of their attributes.

* Corresponding author. Tel.: +972747375111.
E-mail addresses: eitanme@il.ibm.com, eitanme@post.bgu.ac.il (E. Menahem), alonschc@mta.ac.il (A. Schclar), liorrk@post.bgu.ac.il (L. Rokach), elovici@post.bgu.ac.il (Y. Elovici).

http://dx.doi.org/10.1016/j.ins.2015.07.007
0020-0255/

1.1. XML anomalies

Anomalies are data patterns which are either very rare or novel. In the scope of this paper, anomalous patterns are defined according to the structure and the content of an XML document. Such patterns can be generated by many sources, e.g., malicious cyber-attacks or simple typos. Next, we describe the two most prominent anomalous pattern generators.

XML attacks

Applications that interact using XML messages, such as various Web-services, are exposed to a wide range of malicious attacks. These attacks exploit various vulnerabilities in the XML processing mechanism, for example, soft spots of XML parsers or weaknesses of input verification procedures in the target server application. Most common XML attacks include: input validation attacks [28]; probing [40]; malware infiltration; buffer overflow [28,40]; XML parameter poisoning [37,40]; CDATA field attacks [37,40]; SQL injection [28,37,40]; cross-site scripting [28]; schema poisoning [25]; denial of service (DoS); Distributed DoS; XML bombardment; DOM parser DoS attacks; XML Bomb [38] and repetition attacks. These XML attacks usually produce XML anomalies since they appear as string expressions (or by other data types) that are very unlikely to occur with respect to the most common XML documents in their domain.

Data leakage poses another threat to modern information systems. Common causes of data leakage include Trojan attacks, SQL injection attacks, and simple human errors. There are many ways in which outgoing XML documents can lead to data leaks in the system. The simplest way results from putting all the data as it is into a single field that is not properly constrained by a regular expression.

Benign anomalies

Not all XML anomalies are the result of a cyber-attack or a malicious action. Benign anomalous XML documents may be caused by various reasons, such as: honest user mistakes, application errors and communication errors, to name a few.

1.2. Problem statement and applicability

This work focuses on the problem of detecting and localizing anomalies in readable XML documents at computer endpoints. It does not address XML parser attacks since they can be efficiently detected at the network level by technologies, such as *Anagram* [41].

The algorithms which are presented in this paper aim to detect anomalies that result from either malicious or benign actions. We detect both types of anomalies since both of them can inflict damage on information systems.

We would like to stress that the present work does not try to infer the nature of the detected anomalies as this would require a forensic investigation and an understanding of the data domain and semantics. We use *XML-AD* only as an indicator for what could be a network or system attack, which is being delivered by XML documents. As such, *XML-AD* can be applicable, for example, for endpoint anomaly-based XML-Firewalls.

1.3. Anomaly detection

Anomaly detection [1–3,22,24,27,31,43,44] is a process aimed at discovering patterns in datasets that deviate from the common patterns or the expected behavior of the majority of the data. Anomaly detection can be found in a broad spectrum of applications such as intrusion detection, cyber-security, fraud detection, financial systems, and military surveillance, to name a few. Anomaly detection methods employ a wide range of techniques that are based on statistics, classification, clustering, nearest neighbor search, information theory and spectral analysis.

There is an infinite number of anomalous patterns, some of which are very rare and hard to obtain. Consequently, supervised-learning is impractical since training a supervised classifier demands at least a single example from each of the patterns that must be classified. Moreover, in many real-life domains, *normal* examples are inherently easier to obtain than anomalous ones. In such domains, the semi-supervised anomaly detection approach (also known as one-class learning) is employed since it only requires normal class patterns [22]. Anomlay detection of XML documents falls into this category since at the time of the training only normal XML document examples are available.

1.4. Similarity-based anomaly detection

In order to classify a new XML document, x_{new}, as *normal* or as *anomalous*, most anomaly detection algorithms measure $S(x_{new}, X^{normal})$ which is the similarity between the new document and a set of XML documents, X^{normal}, that are known to be *normal*. If the new document is similar to the *normal* instances, it is labeled as *normal*; otherwise, it is labeled as *anomalous*. A common similarity score calculates the distance from x_{new} to the closest *normal* XML document in X^{normal}, i.e., $S(x_{new}, X) = argmin_{x \in X^{normal}} d(x_{new}, x)$, where $d(\cdot)$ denotes a distance function. Finding a distance function which is able to distinguish between the *normal* documents and the *anomalous* ones in the feature space is a fundamental challenge in XML anomaly detection [26].

Most similarity-based anomaly detection algorithms use a multivariate distance function. However, such functions are susceptible to the *curse of dimensionality* [4], namely that they do not reliably measure similarity of high-dimensional data due to