International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 4, No. 2, 2017, pp. 198-209.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

A Novel Approach for the Detection of Anomalous User Behavior in Web Applications Processes (Using Web Applications Firewall- WAF)

Majid Bakhshandeh Nejad[a], Hamid Reza Shahriary[b]

[a] *Science and Research Unit of Islamic Azad University of Qazvin. Iran*.

[b] *AmirKabir University of Technology, Tehran. Iran.*

Abstract

The main challenges of web based systems security would be the detection of sabotage, attacks and anomalous user activities. Those activities threaten the integrity, confidentiality and the availability, of system resources. Sometimes, those activities might seem harmless and normal but considering different factors such as time, address, and the nature of applied changes and also order analysis would uncover the anomalous and malicious nature of them. Most web based systems consist of distributed characteristics and have various stages with significant order, and also use a service related user ID. In this paper, we have presented a method for detecting anomalous user activities, our method consist of analyzing the 3 factors of time, address and request order simultaneously for each user requests in order to make the best decision based on the analysis results. The suggested method works in 2 stages of training and detection. In the training stage, the system will be trained in order to create the required features for the analysis, and the analysis outcomes are used in the detection stage for the decision making process. In order to create diversity in the technology used in web bases systems and avoiding method dependency on one certain technology for easy use in every web based system, we designed our method based on the chronology of web based applications in security tools. The method was implemented on the firewall of an open text application in a web based application by means of PHP language. It is worth mentioning that we have considered different scenarios of anomalous user activities for the implementation processes.

Keywords: Anomaly; Web based system; Applications firewall; Users

Introduction

One of the main challenges in the field of security is related to the diversity and development of system intrusion operations, which threatens the integrity, confidentiality and availability of system resources and network. Even though different measures and strategies have been taken to fight against the attackers, but they seem to come up with new methods every day. Intrusions are done by means of various methods, sometimes they are in the form of malwares which consist of worms and viruses which intent to change or delete information, and sometimes they are in the forms of software attacks carried out by hackers, this one might involve various and diverse methods and tools. These factors accentuate the need for detection and prevention systems.

In general, anomalous activities detection and prevention systems work in two signature based and anomaly based forms: in the first approach which is also known as misuse activities detection scheme, the system identifies anomalous behaviors and takes a sample from the malware behavior as the malware signature, therefore system intrusions will be discovered when they would match with one of the malware signatures samples. In the second approach which is also known as anomaly detection scheme, the system models normal system behaviors and discovers system intrusions based on the normal model.

Identifying those anomalous activities whose behavior patterns are recognizable for the system is rather easy and accurate, but in case of new intruding behaviors the normal behavior modeling scheme is necessary for the detection of anomalous activities. Another circumstance is also imaginable for the second scheme which is based on the dependency of normal behaviors to varied parameters such as: users' working behavior which might make the anomaly detection process rather difficult.

In recent years the popularity of web applications had grown tremendously, and they are being used in sensitive security areas such as: medical, financial and military systems. However, more use of them in sensitive fields and areas might bring numerous and complex attacks against them as well. The majority of web based attacks detection methods involve analyzing the web based application's interaction with customers and supporting servers. Although those methods might be able to identify and prevent some attacks effectively, but they are unable to identify external behaviors of web based applications.

In this paper, we have presented a method which uses web application firewalls, various parameters of users' sent requests and anomaly detection algorithms along with adequate and significant rules of firewalls in order to detect anomalous user behavior in web based application processes.

Anomaly and Classification

Even though the concept of anomaly has been defined many times, but it can be simply defined as the deviations from a normal behavior in the patterns of a dataset (Ghorbani, Lu, 2010). In data mining, anomaly detection or outlier point detection refers to the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Figure 1 shows the anomalies of a simple 2 dimensional dataset. Since most data observations are located in 2 areas of N1 and N2 thus it can be said that all normal data are located in those two areas. Therefore, other areas which are sufficiently distant from those areas can be identified as anomalies. In other words, points of 01 and 02 along with points located in area 3 are considered to be anomalies (Chandola, Banerjee, Kumar, 2009).

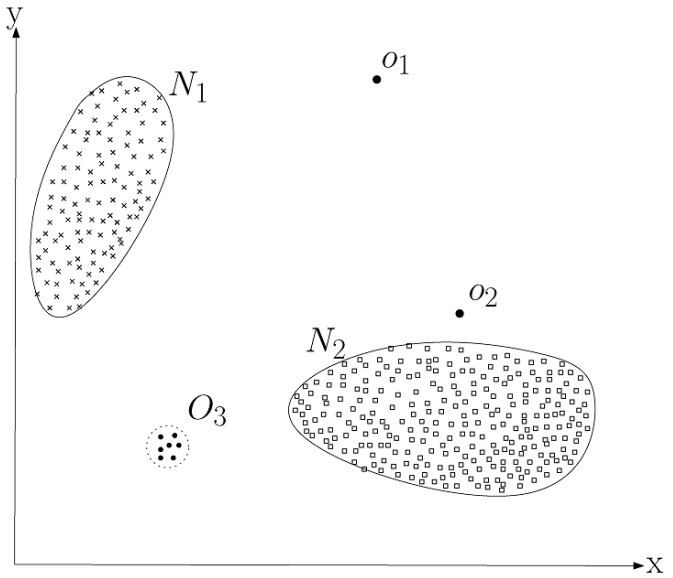


Figure 1: Anomaly sample in a 2 dimensional dataset

One of the main aspects of anomaly detection schemes involves the desirable structure and nature of an anomaly. Anomalies are classified into the following categories (Chandola, Banerjee, Kumar, 2009):

- Point Anomaly

When an individual dataset deviates from other items in a dataset, that sample data can be considered as a point anomaly. This is the simplest type of anomaly, and most anomaly detection studies are focused on it.

- Contextual Anomaly

When a sample data deviates from others in terms of a particular topic or theme it can be considered as a contextual anomaly (SONG, WU, RANKA, 2007).

- Collective Anomaly

When a dataset of related data deviate from other datasets they can be considered as collective anomalies. In a collective anomaly, individual datasets are not anomalies by themselves but when put together with other datasets their deviations from the whole set would be noticeable.

The classification of anomaly detection can be done based on various parameters. In their study Varun Chandola and his colleagues, have suggested various methods for anomaly detection classifications, which is presented in figure 2 (Chandola, Banerjee, Kumar, 2009):

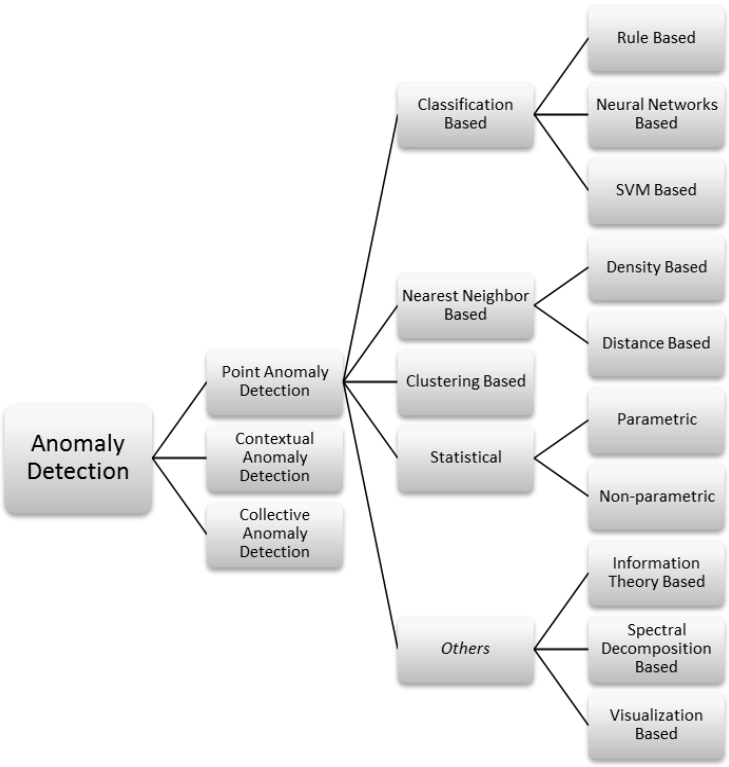


Figure 2: Categorization of intrusion detection methods

Sequence Based Anomalies

A sequence can be defined as a particular order in which things follow each other. Sequences have different types of: binary, discreet and consecutive based on the particular event that forms them. Discreet and consecutive (or time series) sequences have the most real life applications. Discreet / symbolic sequences are sets of orderly events which in fact are symbols of a finite alphabet. For instance, a written document would be a sequence of words as a computer program would be the sequences of system requests, just like a gene is a sequence of deoxyribonucleic acids (DNAs).

The followings are two types of formulas for anomaly detection in consecutive sequences: (Chandola, Banerjee, 2012):

- Semi Supervised Anomaly Detection:

Semi supervised anomaly detection techniques involve constructing a model which would represent normal behavior from a given normal training dataset, and then testing the likelihood of a test instance to be generated by the learnt model.

- Unsupervised Anomaly Detection:

Unsupervised anomaly detection techniques detect anomalies in an unlabeled test dataset.

Based on former analysis sequence anomaly detection would be classified into 3 distinct categories:

- Based on sequence
- Based on repeated sub sequence
- Based on frequency pattern

The first formula indentifies anomalous sequences based on normal sequences. The second formula identifies anomalous subsequences within a long repeated sequence, in other words it involves looking for deviations in subsequences from other subsequences within the long repeated sequence. Finally, the third formula identifies a pattern in a test sequence whose frequency of occurrence deviates from its rate in a normal sequence.

Literature Review

This section analyzes other techniques, studies and algorithms used for detection of anomalous user behavior in web-based applications. Kruegel and Vigna's groundbreaking paper that was published in 2003, suggested the very first anomaly detection system of web based attacks. Mimosa and Waller's models which present two methods for process vulnerabilities analysis by means of source codes are also worthy of recognition (Balzarotti, Cova, Felmetsger, Vigna,2007), (Balzarotti, Cova, Vigna, Cavedon,2010).

- SAD: web session anomaly detection based on parameter estimation :

The above technique detects web attacks without having access to attacks signature. This method involves the development of a normal usage profile and its comparison with the frequency of web accesses logs, in order to ensure a safe web service (Cho, Cha 2004).

- Swaddler : an approach for the anomaly detection of intended workflow violations

The above technique presents a novel approach to the anomaly detections of attacks against web applications. Swaddler presents an accurate description of the internal state of a web application by means of abnormal models. Through request analysis, Swaddler is also able to identify attacks such as violations of the intended workflow of a web application (Balzarotti, Cova, Felmetsger, Vigna, 2007).

- Block technique: a black box approach for detection of workflow violation attacks

Since application logic flaws are specific to the intended functionality of a particular web application it is difficult to develop a general approach that addresses state violation attacks. To date, existing approaches all require web application source code for analysis or instrumentation in order to detect state violations. Block (black-box approach) discovers application logic flaws and allows restrictive functions and sensitive information to be accessed at inappropriate states. This technique uses a system prototype based on the application proxy and extracts a set of request/response sequences and their associated session variable values during its attack-free execution. The set of invariants is then used for evaluating web requests and responses at runtime (Li, Xue, 2012).

- Detection of anomalous user behavior in web application workflows by means of a hidden Markov Model (HMM)

This method detects anomalous user behaviors based on the sequence of their requests within a web session. At first web sessions are decomposed into workflows based on their data objects. Next, a hidden Markov model (HMM) is used in order to characterize workflows on a per-object basis. In this model, the

implicit business logic involved in this object defines the unobserved states of the Markov process, where the web requests are observations. In order to derive more robust HMMs, the object-specific approach would be extended to an object-cluster approach, where objects with similar workflows are clustered and HMM models are derived on a per-cluster basis. This approach consists of two user behavior and attack models (Li, Xue, Malin, 2013).

- Negative security model for anomaly detection of web applications

The negative security model provides a Web Application Firewall (WAF) engine with a rule set in order to ensure critical protection across web architecture. WAFs are deployed to establish an increased external security layer to detect or prevent attacks before they reach web application (Auxilia, Tamilselvan, 2011).

Table 1, compares the efficiency of some characteristics of the above methods.

A1: Swaddler model
A2: SAD model
A3: Block model
A4: Based on hidden Markov Model (HHM)
A5: Based on negative security model

Table1: a comparison of anomalous user behavior detection methods

Type of usage	Location Parameter	Time Parameter	Other Attacks	State violation attacks	Application Logic Attacks	Code Dependency	Technique
Internal	-	-	-	*	*	*	A1
Internal	-	-	-	*	*	-	A2
Proxy	-	-	-	*	*	-	A3
Internal	-	-	*	*	*	-	A4
Proxy	-	-	*	-	-	-	A5

Applications Challenges of the suggested Method

In order to provide an improved and speedier service delivery, huge organizations have started to offer their service through web and web applications. The characteristics of those services are as follows with this reminder that the mentioned software should be compatible with them:

1. They are distributed and available across all geographical locations (Address)
2. Those services follow distinct stages whose order must be kept at all times.(Stages)
3. Services are based on distinct IDs which are the criteria for web applications.(ID)
4. They all follow a specific time pattern. (Time)

Most service provider applications treat these characteristics as challenges of the project.

Model of the Suggested Method

Since we have considered the factors of address, stage, ID and time as the challenges of service provider applications, thus our suggested model of the project is based on the combination of multiple system request state. Figure 3 presents an overall model of our suggested method.

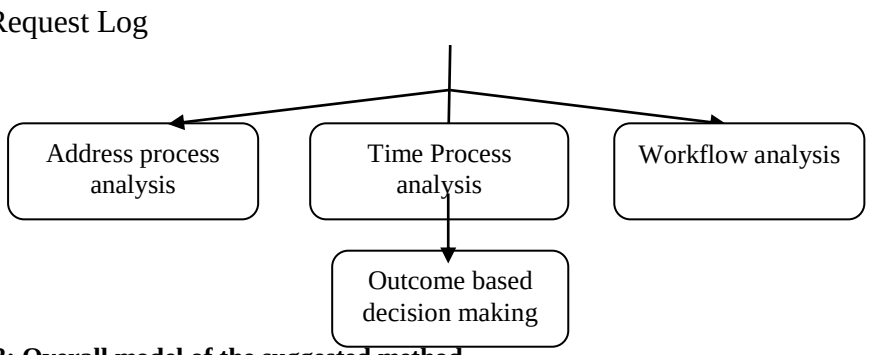


Figure 3: Overall model of the suggested method

The above method will be carried out in two stages of training and detection. In the training stage, system learns the analysis methods and the related profiles are being created. In the detection stage the system uses the analysis outcomes in the decision making process.

Structure of the Suggested Method

After discussing the model and structure of our suggested method, the following sections clarify its way of implementation and different aspects of it.

The various stages of the above structure are as follows:

1. A proxy receives the requests and sends them for pre-processing
2. Various stages of pre-processing and request analysis are as follows:

- Omitting the flawed , useless, etc requests
- Request decomposition and required field specification such as: user's IP address, web sessions ID, User ID in web application, sent parameters, time and other parameters.
- Sending the information to " processing engine" and " learning system"
- "Rules Database" section manually defines the conditions for each factor (time, location and process workflow).
- Decision management section, sends the extracted information to 3 sections of time process, address process and workflow process analysis. Next, those 3 sections use the "rules database" information for processing in order to create an adequate ticket/alarm or reaction in the system.

Compliance Method and Anomaly Detection

This project is based on two stages of training and basic data analysis, to do so first we cluster training data or algorithms, and then we calculate the distance to the center of nearest clusters for the test data. Therefore the normal data can be distinguished from its anomalous counterpart based on those distance and threshold calculations. We have used a script execute function for anomaly detection in which we put the related code for the test data. Figure 5 presets an approximate model of the mentioned code.

/*Workflow Anomaly Detection Procedure*/

Input: A set of Cluster {C_i} From Training Data and User Scenario Test Data{Ud_i};
/*
We need Cluster {Ci} Set Of Training Data and of User Scenario Test Data{Ud_i}
To Calculate Anomaly Outlier Score
*/
Output: Outlier Score For User Scenario Test Data{Anomaly-Ud_i};
Def Set Of Cluster {C_i};
Def Set Of User Senario Test Data {Ud_i};
While Record (R_i) in User Scenario Test Data {Ud_i} **do**
for i **to** length os Cluster {C_i} **do**
Calculated Outlier Score For User Test Data ;
/*
Outlier Score Calculation With The distance
To the Center Of nearest Cluster
*/
Set Max of Outlier between Clusters {C_i};
End

End
Return User Scenario Test Data{Ud_i} with New Column (**Outlier**);

Figure 5: The approximate model of anomaly detection code

We have used the above algorithm for each 3 sections and the analysis output for each was sent to decision management. Finally, the final decision is made based on the risk level. The decision management makes its decision based on the analysis of each 3 factors, but it is worth mentioning that considering one stage or stages of user activities and requests could make the anomaly detection process error-free and more accurate. Equation 1 presents compliance method and anomaly detection procedure for this section.

$$1)\quad Final = (TXi + IXi + WXi) \begin{cases} >0 & /* \text{ Anomaly} \\ =0 & /* \text{ normal} \end{cases}$$

$$2)\quad Final = \sum_{i=0}^{n}(TXi + IXi + WXi) \begin{cases} >0 & /* \text{ Anomaly} \\ =0 & /* \text{ normal} \end{cases}$$

Equation 1: Management section calculations

The formulas TXI, IXI, WXI have been used for time, address and workflow calculations respectively.

Implementation

Our suggested method involves two procedures of receiving user requests and analyzing their parameters. Those procedures were carried out by means of web application firewalls (WAF). Web application firewalls control all requests in the outlier layer in order to analyze and prevent attacks. A firewall of a web application can be exploited as an independent web server plug-in device and is able to impose a set of rules on web protocols.

Mod-security is the name of a popular layer application firewall which works as an additional module for Apache web servers. By installing mod-security in reverse proxy-mode you'll be able to protect any number of web servers and monitor all the organization's services regardless of their programming language or applied technologies.

What is created and sent for pre processing here would be the related log to the received request. A sample of that log is presented in figure 6.

[08/Aug/2014:01:05:46+0300]
[2.168.16.112/sid#28a6ae10][rid#29a06058][/WAdmin/mg_pgs/main/Frm_Chg_List.php][1] Access denied with code 500 (phase 3). Pattern match ".*" at ARGS_GET:tp. [file "/usr/local/etc/apache22/Includes/mod_security2/activated_rules/whitelistip.conf"] [line "41"] [id "10101"] [msg "Get data: ARGS_GET:tp=Attack"]

Figure 6: Log and output sample of a request receiving stage

The above method was implemented and tested on a web based assessment and monitoring system created with PHP language. Finally, we built a model of the extracted data by means of Rapid Miner software and K-Medoids and K-means algorithms. Details and other applications of the mentioned algorithms are provided in: (Rastegari and Mobin, 2016) and (Mobin et al., 2017). To do so we'd considered 3 scenarios and tested them in a data training model (consisting of over 4500 records-over 1000 test data).

Results for the implementation of this method on the above data are presented in tables 1, 2 and 3.

- User exploitation of system from authorized addresses in improper times (improper for sending requests)
- User exploitation of system in proper times from authorized addresses
- User exploitation of system (proper time – unauthorized address) outside of system accessibility area (accessibility level violation)

Table 2: The results for K-Medoids and K-means algorithms implementation in 1st scenario

Scenario 1										
		Test Data Count	K-Means				K-Medoids			
			TP	FP	TN	FN	TP	FP	TN	FN
User1		1039	148	0	891	0	148	0	891	0
User2		1020	154	0	865	1	155	0	865	0
User3		886	87	19	769	11	94	3	785	4
User4		904	115	0	786	3	115	0	786	3
User5		934	97	0	835	2	98	0	835	1
Specificity	Sensitivity		97.2%		99.5%		98.6%		99.8%	

Table 3: The results for K-Medoids and K-means algorithms implementation in the 2nd scenario

Scenario 2										
		Test Data Count	K-Means				K-Medoids			
			TP	FP	TN	FN	TP	FP	TN	FN
User1		1039	48	0	991	0	48	0	991	0
User2		1020	40	0	970	10	40	0	970	10
User3		886	25	12	874	0	25	12	874	0
User4		904	22	0	882	0	22	0	882	0
User5		934	50	0	884	0	50	0	884	0
Specificity	Sensitivity		94.8%		99.7%		94.8%		99.7%	

Table 4: The results for K-Medoids and K-means algorithms implementation in the 3rd scenario

Scenario 3										
		Test Data Count	K-Means				K-Medoids			
			TP	FP	TN	FN	TP	FP	TN	FN
User1		1039	8	0	1027	4	1	0	1027	11
User2		1020	11	0	1006	3	4	0	1006	10
User3		886	8	0	875	3	7	0	875	4
User4		904	7	0	804	3	6	0	804	4
User5		934	11	26	893	4	9	0	919	6
Specificity	Sensitivity		73.5%		99.4%		44.5%		100%	

Conclusions

By considering the results of our implemented method, we conclude that user workflow process in web applications is full of valuable details which can be useful for the anomaly detection procedure. Moreover, combining factors such as request receive time and the distribution in system requests might increase the accuracy of the mentioned process and may also reduce the duration of decision making. Since, the diversity of programming languages and technologies used in systems might make the exploitation of anomaly detection methods rather difficult; therefore by using web application firewalls as means of method implementation we've eliminated that problem effectively.

We have also found that using this method along with other user processes effective parameters might optimize the anomaly detection procedure. Finally, the detection procedure can be speedier based on each parameter of an algorithm.

Future Works

The presented method basically accentuates the importance of factors such as time, distribution and process stages which can make for an easier anomaly detection procedure. Since the detection of anomalous behaviors of authorized users is proved to be difficult, we have suggested combining non-compliance of a user request with characteristic such as time, distribution and stage along with the use K-Medoids and K-Means algorithms to solve that issue, which had remarkable outcomes. Therefore, the development stages should consider the following factors:

- Using combined form of the above factors with 1% effectiveness coefficient, in other words combining risk management in the above method in order to provide a better condition for decision making
- Using artificial intelligence algorithms for each stages of the above system for effective exploitation
- Using the above method along with other security systems such as intrusion respond systems

References

Auxilia.M, Tamilselvan.D. (2011). *Anomaly Detection Using Negative Security Model in Web Application. Computer Information Systems and Industrial Management Applications (CISIM),* IEEE.481-486.

D. Balzarotti, M. Cova, V. V. Felmetsger, and G. Vigna. . (2007). *Multi-module vulnerability analysis of web-based applications. InCCS'07: Proceedings of the 14th ACM conference on Computer and communications security*. 25–35.

Ghorbani, W. Lu, and M. Tavallaee, (2010). *Network intrusion detection and prevention: concepts and techniques* .vol. 47: Springer

M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna. (2007). Swaddler: *An Approach for the Anomaly-based Detection of State Violations in Web Applications, in RAID'07: Proc. of the 10th International Symposium on Recent Advances in Intrusion Detection*. 63–86

Mobin M., Li Z., Komaki M., A Multi-Objective Approach for Multi-Stage Reliability Growth Planning by Considering the Timing of New Technologies Introduction, IEEE Transaction on Reliability, DOI: 10.1109/TR.2016.2638124, 2017.

S .Cho, S .Cha , (2004), *SAD: web session anomaly detection based on parameter estimation. Computers & Security*,vol 23.. 312-319

Song, X., Wu, M., Jermaine, C., and Ranka, S. 2007. *Conditional anomaly detection.* IEEE Trans. Knowl. Data Eng. Vol .19, 631–645.

V. Chandola, A.Banerjee, V.Kumar. (2009). *Anomaly Detection: A Survey . ACM Computing Surveys*, Vol. 41. No. 3. 1-72.

V. Chandola, A.Banerjee .(2012). *Anomaly Detection for Discrete Sequences: A Survey* .IEEE transactions on knowledge and data engineering, Vol. 24. 823-839

V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna. (2010). *Toward Automated Detection of Logic Vulnerabilities in Web Applications. In USENIX'10: Proceedings of the 19th conference on USENIX Security Symposium*. 143–160

Xiaowei Li ,Yuan Xue, Bradley Malin. (2013). *Detecting Anomalous User Behaviors in Workflow-driven Web Applications. Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium*.1-10.

Xiaowei Li, Yuan Xue. (2012) .*BLOCK: A Black-box Approach for Detection of State Violation Attacks Towards Web Applications*, ACSAC. 247-256

Rastegari A., Mobin M., Maintenance Decision Making Supported by Computerized Maintenance Management System, Proceeding of 2016 Annual Reliability and Maintainability Symposium (RAMS 2016), Arizona, USA.