A Data Mining Based Approach for Mitigating Attacks on Web Services

Sunil Sharma[1], Mohit Kumar[2], Kalka Dubey[3]

[1]Assistant Manager IT, MPPKVVCL(MPSEB), Indore
[2]Assistant Professor, JP Institute of Engineering & Technology, Meerut
[3]Software Developer, Aricent Group, Gurgaon

Abstract—*Web-Services are very crucial in today's web based business infrastructure. They are able to dilate the way on which business runs. Web-services make services accessible from anywhere and can transform applications to web applications. They are basis of many B2B communications. Due to their effect on business they woo attackers to disrupt the service provided by different means. The objective of this paper is mainly attack avoidance based on data mining. We are presenting a prototype of model for attack avoidance on web-services. The model can intelligently avoid some types of attacks on web-services. The aim here is to rectify avoidance probabilities of attacks and decrease false positives and false negatives.*

Keywords—KD algorithm, web-service security, SOAP, SOA.

I. INTRODUCTION

Web Services and Service-Oriented Architectures (SOAs) are often considered to be among the most important technological innovations of the last decade [1]. With the magnificent popularity and growth of the Internet, numerous web applications and web services are being deployed. Web services are self contained components of software's that are meant to be providing services to other Internet users. Web services are widely used in business transactions and are meant to be used business to business (B2B) model. Web-service is mainly based on Extensible Markup Language (XML) and Hyper Text Transfer Protocol (HTTP). Communication protocol for web-services is Simple Object Access Protocol (SOAP). SOAP messages are transported using HTTP, SMTP etc. Web-services are described by Web Service Description Language (WSDL). We use Universal Description, Discovery and Integration (UDDI) to locate and access a service via service metadata.

Main feature of web-services is that they provide many open standards and interoperable architecture. Using web-services our legacy applications can communicate to other applications developed and operating on entirely different platforms. Service Oriented Architecture (SOA) is a methodology for development of software in terms of interoperable and self contained services. Services are linked through well defined interfaces. The interfaces are defined in neutral manner and should be independent of implemented services, hardware platforms, operating systems and programming languages. It makes the services interact in a uniform and universal way which are built in a variety of systems [2]. As SOA is accepted to be a popular model for software development it is not free from attacks. Services communicate via SOAP messages. The SOAP server usually runs on a web-server. Therefore the threats existing for a web-server also exist for a SOAP server [3].

As web services are adopted to be new revolution in businesses, attackers are wooed to hack web services and provider of those services. The main requirements of any secure system are integrity, confidentiality and availability. Any attempt to adversely affect these requirements can be termed an attack. These attacks can be launched on a system whose vulnerabilities can be exploited. Any web service provider hosting a business critical web-service must ensure the security of the system. In the development of Web service's logic functionality, if the security features are designed, Web services will become extremely complex; and the service performance and scalability will be greatly reduced [2]. There exist some standards and specifications such as WS-Security[4],WS-Secure Conversation[5], WS Security Policy [6] etc. which are meant for applying security during communication of the SOAP messages and usage of web services.

This paper discusses different attacks on web-services and proposes a prototype for securing web-services against some of possible attacks in different ways using data mining. Security related task is separated from logic function due to complexity and performance issues.

Rest of the paper is organized as follows- Related work has been discussed in section II. In section III web service is explained. In section IV some attacks on web-service are discussed. Proposed model is explained in section V. Experimental set up for model simulation is discussed in section VI. In section VII two algorithms of clustering namely K-means clustering and KD algorithm have been discussed. Conclusion is presented in section VIII.

II. RELATED WORK

In the related security implementation, two techniques have been discussed, viz. anomaly detection and signature detection or misuse detection. Anomaly detectors identify abnormal behavior patterns. They are based on assumption that attacks are different from normal activity and therefore can be detected by a system that can identify these differences. The system establishes a baseline of normal usage patterns and anything that widely deviates from it is considered to be a possible attack. The baseline is constructed from historical data collected over a period of normal operation.

Misuse detectors or signature detectors analyze system activity, looking for events or set of events that match a predefined pattern (signatures) of events that describes a known attack [7]. A tool called HoneyAnalyzer for signature extraction as discussed in [8] does not consider attacks at

Application layer. Machine learning classifiers use object characteristics to identify the class or group it belongs to. A linear classifier achieves classification based on the value of a linear combination of characteristics that are presented in the form of a vector ’feature vector’ to the machine for classification. Several classifiers exist such as Neural network based [9], Genetic Algorithm based [10], Decision tree based [11] etc. These have been used for classification of intrusion attacks by various researchers [12][13]. The data sets used in these works contain intrusion attacks at network and transport layer.

The intrusion detection based on data mining plays an important role. Such methods based on some specific algorithms of data mining have been implemented for intrusion detection at network or transport layers. The work most similar to unsupervised model generation is a technique developed at SRI in the Emerald system [14]. For building normal detection models and comparing distributions of new instances to historical distributions Emerald uses historical records. Related to automatic model generation adaptive intrusion detection, Teng et al. [15] do real time anomaly detection by using inductively developed sequential patterns. Sobirey’s work on adaptive intrusion detection using an expert system to collect data from audit sources is also relevant [16]. In the data mining and machine learning, several such methods were discussed but most of them implemented for network and transport layer. Application layer security for web-services seldom discussed. We are proposing such system for security of web-services.

III. WEB SERVICE

A web service as defined by W3C is a software system designed to support interoperable machine to machine interaction over network. A web service is identified by a Uniform Resource Identifier (URI) whose interfaces and bindings can be defined, described and discovered as XML artifact. A web service is an application component which communicates using open protocols and is self contained and self describing. They play an active role in the business integration due to interoperability and loosely coupled messaging framework. Three main elements of web services are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), Universal Description, Discovery and Integration (UDDI).

SOAP is a W3C standard for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP is important for application development as it allows Internet communication between two or more programs. Furthermore, SOAP is platform independent and general purpose XML protocol [17]. WSDL is an XML based language used to define web-services and describe how to access them. Various operations required to access web services are defined in WSDL along with the parameter information [17]. UDDI is a specification for publishing and locating information about Web services.

IV. ATTACKS ON WEB-SERVICES

No system is perfect and there may exist vulnerabilities in the system which can be exploited. SOAP and XML are no exceptions. Due to this web services are vulnerable to attacks. Some common attacks targeting the Web services [1][18] are discussed below.

(1) Denial of Service Attacks

This is a direct attack on availability and also the most common attack which can be launched in several ways. When this attack is launched, the server hosting the service runs out of resources and cannot respond to legitimate consumers. Coercive parsing and oversized payload are examples of this kind of attack. Parsers based upon document object model are very susceptible to this attack as they represent entire XML document in memory. Server may take very long time to respond to attack requests as parser consumes time to parse whole request, thus may not be available for legitimate clients. Performing DoS attack an attacker may attack a router, firewall, or proxy server with the aim to make them unusable. In short the DoS attack can make services unavailable.

(2) Injection Attacks

In this kind of attack an unexpected string which contains some executable logic or XML tags are inserted in request message to cause the undesired effect on server. By using an SQL injection attack a web-service enabled database application can be targeted to reveal unauthorized information or destruct information. In XML injection attacker can insert XML tags in such a way that cannot be detected by parser and may lead to undesired effect, e.g. parameter overriding. Attacks like XSS can also be launched on web-services. Characters like “<” and “&” are illegal when used as a content of s ome element. Someone who wants to insert a script in a message can insert it into <script> tag. To avoid unwanted parsing errors, script code is defined as CDATA. Everything inside CDATA section is ignored by the XML parser. This may lead the attacker to insert a script which can harm system operation.

(3) Oversized Payloads

This attack like coercive parsing is a DoS attack in which attacker sends a very large formulated message to server to cause resource exhaustion. This is due to the reason that document object model based parser bring entire message to memory so server may run out of memory.

(4) Principal Spoofing

This is a kind of attack on authentication and authorization. In this attack, attacker sends spoofed message to receiver and deceive receiver to believe that it is from a legitimate sender. Example of this attack is IP spoofing in which attacker spoofs a valid IP address and pretends to be valid user. If he gains access to system he can disrupt the normal operation of system.

(5) Buffer Overflow Exploits

Buffer overflow exploits the target web-services which are implemented in a language that is vulnerable to buffer overflow such as C or C++. These attacks may be very dangerous and may cause entire system to crash. When data is stored in memory, a memory area is allocated to those data. If data is very large and service fails to check the boundaries of allocated buffer, than buffer overflow occurs and system may crash.

(6) Schema Poisoning

XML schemas are standard of a web-service request which every request must follow in order to be proved valid request. In addition they also instruct the parser how to parse the message. Service consumer access schema and according to that form the message. Schemas are located and stored at some location. If schema is compromised it can be replaced or modified by an attacker. Now request to web-service on previous schema are no more valid and cause the service to unavailable or accept malicious request now on.

(7) Registry Disclosure Attacks

Registry disclosure attacks are caused by mis-configured registries such as LDAP, X.500 etc. These registries contain sensitive information regarding authentication and about web-service etc. so if registries are compromised or corrupted they may lead to reveal information about web-service’s host or even gain access to that host.

(8) SOAPAction spoofing

Original operation addressed by a SOAP request is identified by the first child element of the SOAP body element. Also HTTP optional header field SOAPAction can also be used to intended operation requested by SOAP message. This SOAPAction field can be used by web-services to identify the operation to get rid of XML processing for optimization. This can lead to man-in-the-middle attack in which an attacker changes the operation in SOAPAction to other operation that is not similar to the SOAP message. So the server responds with undesired result.

In addition to these attacks there are many other attacks like attack through SOAP attachment, XML re-writing attack, format string attack etc. Each attack is exploiting some vulnerability in SOA. Here we make an attempt to avoid some of these attacks.

V. PROPOSED MODEL

The proposed model is shown in Fig. 1. As we can see it works on application layer and implements security mechanisms using data mining. It is assumed that authentication has been done. Whenever a SOAP request arrives it is first transferred to data preprocessor which separates meaningful data and useful data. Meaningful data is not in acceptable mining format so it is fed into data format converter, which turns data into a format that is acceptable in data mining. There exist many techniques which can be used to convert it to required format such as data cleaning, data reduction etc. This formatted data is in turn fed into data mining process controller where the main task is performed, i.e., clustering. During testing and training of data sets, clusters have formed using KD clustering algorithm. The clusters may be any number depending upon the data sets. When algorithm receives training data it divides the data sets into number of clusters depending on their similarity to each other. Data sets belonging to same cluster have higher similarity. Here lies the main idea behind normal cluster and attack clusters.

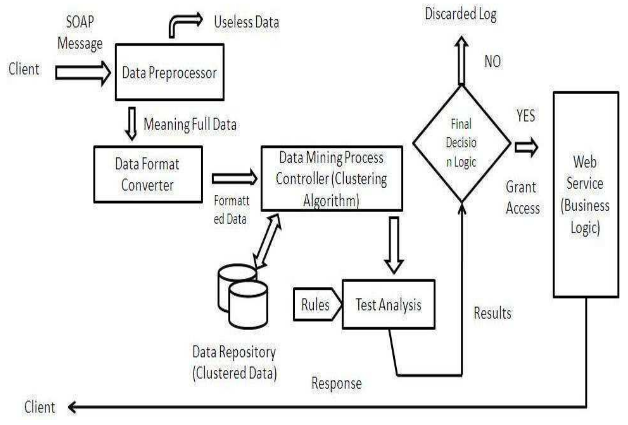

Fig. 1. Proposed model for web-service security

We take advantage of the fact that web-service accepts requests of some well defined schemas and all normal requests always posses that schema. Attack requests are different form normal requests and hence can be classified into different clusters. Different clusters are stored in data repository and any newly arrived request can be compared and/or added to those clusters. Security administrator can define which clusters are normal and which are not. While training, normal data is in majority, It can be assumed that cluster which has higher number of data sets is normal. After comparing to clusters, cluster id is passed to test analysis component which based on some threshold rules can be fed to final decision logic which decides, whether to grant service or not. The discarded requests are logged into discarded log. If access is granted then the request is forwarded to a port where the web-service is running.

This model can be implemented as a separate service, i.e., tester service and can be isolated from business logic to avoid complexity and can be deployed on some other port. When a request reveals to be normal is it forwarded to the port where actual web-service is implemented. Now web-service performs its action and responds to client.

VI. EXPERIMENTAL SET UP

Fig. 2 shows experimental set up to simulate the proposed model.

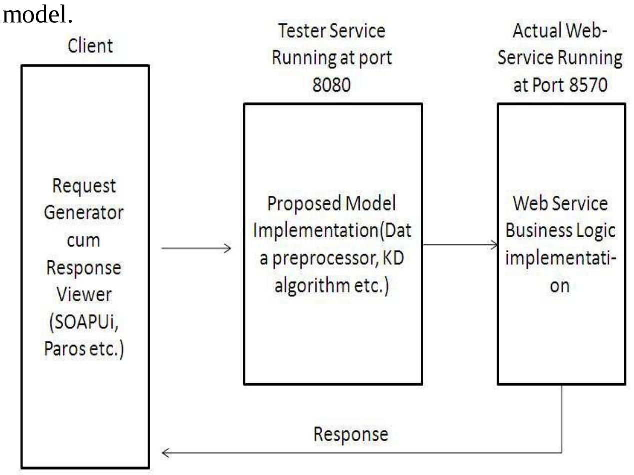
Client
Request Generator cum Response Viewer (SOAPUi, Paros etc.)
Tester Service Running at port 8080
Proposed Model Implementation(Data preprocessor, KD algorithm etc.)
Actual Web-Service Running at Port 8570
Web Service Business Logic implementati-on
Response

Fig. 2. Experimental set up

Here any tool which can be used as a request modifier or generator can act as a client or consumer of web-service. SoapUI is such a tool. SoapUI is an open source web service testing application software for SOA developed by Smart Bear [20]. Request generated are forwarded to a tester service running e.g., at port 8080. Tester service implements the proposed model. If request is found to be legitimate then this tester service forwards the request to another port e.g., port 8570, on which actual web-service is implemented. The response generated from actual web service is returned to the client. Tool acting as client can modify the requests for generating attack requests for target web service. All these requests (normal and attack requests) are used to train the model. We can also analyze the generated response and response time displayed in tool.

VII. K-MEANS CLUSTERING AND KD ALGORITHMS

K-means clustering algorithm is an unsupervised learning method in which task is to divide the given data sets into clusters, data sets in same cluster have higher similarity. It selects k data as initial clusters center and then iteratively compare the distance of data sets from this cluster center. A data set is added to a nearest cluster. After all data sets are processed cluster centers are recalculated and process is repeated until cluster centers don't change. Now data sets are divided into clusters. K-means is simple and traditional algorithm and this algorithm is sensitive to initial value and sequence of data sets. Different initial data sets may lead to different results.

The KD algorithm is another clustering algorithm which is an improved version of K-means algorithm. This algorithm accepts a new parameter called threshold radius R and avoid the need of K i.e., number of clusters in K-means. If a data set is in threshold radius of a cluster it is added to that cluster.

Here is pseudo code of KD algorithm [19], C stands for cluster, dist(C,v) stands for distance of vector v from center of cluster C. Initialize S to null set (Initially cluster set is empty).

```
Pick a vector v from data set.
If (S == null)
{
Ci= Ci(v) //build a new cluster Ci on v.
S++; // increment in number of clusters
go to Step 20;
}
Else
{{
If (dist (Ci, v)<= R)
Ci = Ci +v //add vector v to cluster Ci.
Go to Step 20;
}
Else
{
Ci=new (Cluster) //create a new cluster on v
S++;
}}
```

Repeat step 2 and 3 until all vectors of data set are processed.

Recalculate the center of each cluster, for each cluster scanning data set from initial cluster.

If the distance from certain vector of data set to cluster Center isn't larger than R; add this vector to this cluster.Repeat it until all cluster centers are fixed.

Now we have done clustering and it is the task of security administrator to determine which cluster is normal. Due to the fact that while training, normal data was given immensely normal cluster should be cluster with higher number of data sets. When a request arrives its distance is calculated to different clusters if it is less than or equal to threshold value than it is consider to be belonging to that particular cluster. Now if that cluster has been evaluated to be normal, request is normal otherwise it is considered to be an attack.

VIII. CONCLUSION

Web-services revolutionized the way of building software. We can make services available and reusable by consumers despite of platform considerations. They are interoperable and playing a vital role in business. Due to these reasons they attract attackers to target them. Some attacks on web-services and an attempt to mitigate these attacks are discussed in this paper. We propose model and use of KD algorithm which is an enhanced version of traditional K-means algorithm. Also experimental set up is discussed for simulating the proposed model. This model can be implemented for application layer security of web-services based on data mining.

REFERENCES

[1] Jensen, Meiko and Gruschka, Nils and Herkenhöner, Ralph "A survey of attacks on web services," Computer Science - Research and Development, vol. 24, pp. 185-197, 2009.

[2] H. Yue and X. Tao, \Web services security problem in service-oriented architecture, "Physics Procedia, vol. 24, Part C, no. 0, pp. 1635- 1641, 2012

[3] S. Cable, B. Galbraith et. al., Professional Java Web Services, Shroff Publishers & Distributors Pvt. Ltd.

[4] WS-Security, OASIS identifier wss-v1.1-spec-os-SOAPMessageSecurity,[http://docs.oasisopen.org/wss/v1.1/]

[5] WS-SecureConversation,[http:/docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws secureconversation-1.3-os.html].

[6] WS-SecurityPolicy, OASIS WS-Security Policy specification,[specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf]

[7] Christian Kreibich, Jon Crowcroft, "Honeycomb-Creating Intrusion Detection Signatures" Using Honeypot, ACM SIGCOMM Computer Communication Review archive Volume 34,Issue1 (January 2004), Pp. 51 – 56.

[8] Urjita Thakar, Sudarshan Varma, A.K. Ramani, "HoneyAnalyzer – Analysis and Extraction of Intrusion Detection Patterns & Signatures Using Honeypot", The Second International Conference on Innovations in Information Technology, Dubai, UAE September 26-28, 2005

[9] Shifu Chen, Zhaoqian Chen, F ANNC: Neural Network Classifier" Knowledge fast adaptive and information systems(2000) 2: 115-129.

[10] Eugen Barbu, Romain Raveaux, Herve Locteau, Sebastien Adam, Pierre Heroux, Eric Trupin, "Graph Classification Using Genetic Algorithm and graph probing application to symbol recognition,", vol. -, pp. 296-299, 18thInternational Conference on Pattern Recognition (ICPR'06) Volume 3, 2006.

[11] Anurag Srivastava, Eui-Hong Han, Vipin Kumar and Vineet Singh "Parallel Formulations of Decision-Tree Classification Algorithms", Data Mining and Knowledge Discovery, 3,237-261 (1999).

[12] Elkan, Charles, "Results of the KDD'99 classifier learning", SIGKDD Explorating, 2000, pp.63-64.

[13] Lee W. and Stolfo, S., "A framework for constructing Detection system", features and models for intrusion ACM Transactions on Information and system security, 3 (4), 2000, pp.227-261.

[14] H. S. Javitz and A. Valdes. The nides statistical component: description and justification. In Technical Report, Computer Science Labratory, SRI International, 1993.

[15] H. S. Teng, K. Chen, and S. C. Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. In roceedings of the IEEE Symposium on Research in Security and Privacy, pages 278–284, Oakland CA, May 1990.

[16] M. Sobirey, B. Richter, and M. Konig. The intrusion detection system aid. architecture, and experiences in automated audit analysis. In Proc. of the IFIP TC6 / TC11 International Conference on Communications and Multimedia Security, pages 278 – 290, Essen, Germany, 1996.

[17] Alonso, G. Casati, F. Kuno, H. Machiraju, V, "Web Services:Concepts, Architectures and Applications" , Springer, 2004.

[18] Esmiralda Moradian , and Anne Håkansson, "Possible attacks on XML Web Services", International Journal of Com puter Science and Network Security, Vol.6 No.1B, January 2006, pp154-170.

[19] Haque, M.J.; Magld, K.W.; Hundewale, N., "An intelligent approach for Intrusion Detection based on data mining techniques," Multimedia Computing and Systems (ICMCS), 2012 International Conference on , vol., no., pp.12,16, 10-12 May 2012.

[20] SoapUI,http://smartbear.com/products/qa-tools/web-service-testing-tool.