SOAP: SENSITIVE OPERATIONAL ATTRIBUTE PATTERN BASED VULNERABILITY ANALYSIS FOR BUSINESS INTELLIGENCE USING RULE SETS

[1]S. SENTHIL KUMAR, [2]DR.M.PRABHAKARAN

[1]Research Scholar, Department of Computer Science, Karpagam University, Coimbatore, Tamil Nadu, India

[2]Assistant Professor, Department of Computer Science, Government Arts College, Ariyalur, Tamil Nadu, India

E-mail- [1]senthilphd123@gmail.com

ABSTRACT

The development of internet technology opens the gate for most business solutions to be performed online; for example online shopping, fund transfer, banking process and etc. The problem is, how secure the transactions performed online and what security metrics considered by service providers. By providing online services, the providers of service have great deal of securing user information's and business data from malicious users. There are chances for variety of internet attacks, to degrade the service performance or malfunctioning the user information. Now a day, there is a high frequency of online stealing of user information and anonymous transfer of funds from personal accounts. We propose a naval approach to safeguard the services, information of both personal and corporate from malicious attacks. The proposed SOAP approach which groups services according to the type of attributes they access and how the services are completed. We use service rule sets using which a service access matrix is computed to find out malicious service accesses. Finally we generate set of vulnerability patterns based on service access matrix and rule set.

Key Terms: *Vulnerability, Service Access Matrix, Business Intelligence, Operational Attribute.*

1. INTRODUCTION

The business organizations maintain various information's about the corporate as well as individual or customers. The amount of information is vast in volume and huge in important, because they have the responsibility to maintain all those information for them in secure manner. The kind of risk arises in maintaining the user information's are due to variety of malicious threats which can be generated by dedicated groups or individual. What the threatening groups can do is, they can steal the information and initiate some attacks. For example, if the organization is supporting online shopping, a genuine user will provide his account details like credit card no, to the online interface. So that the card no will be validated through some other interfaces from the organization. Later the user will be asked for transaction passwords and confirmations. What happens when a malicious user catches the secret information is, the malicious user can do some malformed activities and he can shop some goods on behalf of the original user and pay from the steal account. Similarly, a malicious user can perform many ways of threats.

In case of service degradation, a malicious program may use genuine user details to get the service response and processed up to some hard core process and then drop the request. This will hugely degrade the service performance, because the original genuine user will not be serviced. The vulnerability is the term refers to the threat generated and can be measured in many ways. In a transactional service, the vulnerability can be represented by how well the service request is completed and how it gets processed.

The vulnerability has to be measured based on the effect the threat makes to the business system. For example, if a malicious user generates anonymous request through a service provided by the business system, then vulnerability of the service or the user has to be measured based on the frequency of access and how much the service access damages the business service and data. Once the pattern of malicious access or threat has been identified then , the business system has to modify the protocol of access to override the malicious access or vulnerable pattern.

Generally each service has some purpose and internally that will affect the system at minimum level. so that , each service need to access some attributes of the system and can be categorized according to the kind of attribute it access to complete the service. We categorize the attributes as sensitive and non sensitive, also the services are categorized as sensitive and non-sensitive services based on the attributes they access.

2. RELATED WORKS

There are many approaches has been discussed earlier in the literature, we discuss few of the here.

Mitigation of Program Security Vulnerabilities: Approaches and Challenges [6], aims to provide an in depth overview of existing mitigation (testing, static analysis, and monitoring) approaches that are widely applied for program at the “code” and “execution environment” levels. The tutorial focuses on five most common program security vulnerabilities such as buffer overflow, SQL Injection, format string bug, cross site scripting, and cross site request forgery. The tutorial presents existing program security vulnerability mitigation techniques and approaches in terms of their common key characteristics and limitations. Moreover, it also highlights the open issues and future research directions.

Modeling and classification of network vulnerability are introduced firstly, then the concept of attack capability transfer and the algorithm to produce it are presented in [7], which can aggregate vulnerabilities with the same exploitation attributes and satisfying some constrains to simplify the further analysis. Based on the attack capability transfer, a new method constructing attack graph is presented, and the complexity is O(N2) where N is the number of hosts in a network. Through the analysis of attack graph, network vulnerability quantitative analysis is taken and security hardening method based on approximate greedy algorithm is presented.

The Research on Network Vulnerability Analysis Methods [8], summarized the popular vulnerability Analysis Methods in the field of computer network security. This paper also described and compared these methods. Then this paper introduced the Vulnerability analysis methods for communication network, command & control network, mobile ad hoc network and satellite network. At last, the shortage of current research on satellite network vulnerability was analyzed and the next research idea was proposed.

VULCAN: Vulnerability Assessment Framework for Cloud Computing [9], propose a novel vulnerability assessment framework for cloud computing systems. We have designed and developed a prototype of our framework. We also present the design and development of our framework with some use cases.

Ranking Attacks Based on Vulnerability Analysis [12], provide a set of security metrics to rank attacks based on vulnerability analysis. The vulnerability information is retrieved from a vulnerability management ontology, which integrates commonly used standards like CVE, CWE, CVSS, and CAPEC. Among the benefits of ranking attacks through the method proposed here are: a more effective mitigation or prevention of attack patterns against systems, a better foundation to test software products, and a better understanding of vulnerabilities and attacks.

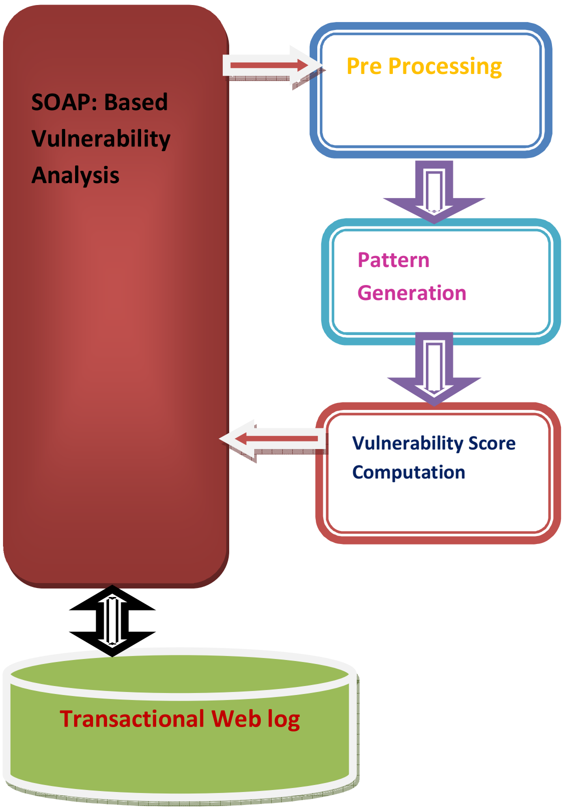
Software vulnerability analysis framework based on uniform intermediate representation [13], presents a static analysis framework based on uniform intermediate representation to detect software vulnerabilities, and we have implemented an analysis tool called Melon based on the Microsoft Phoenix. We evaluate the effectiveness of Melon through a number of testing, and the experimental results show that it can effectively validate and analyze software vulnerabilities.

Structural and functional vulnerability analysis for survivability of Smart Grid and SCADA network under severe emergencies and WMD attacks [14], propose a flexible and extensible framework for network survivability testing and evaluation based interdependency modeling of Smart Grid and SCADA network. We developed novel approach for vulnerability reduction considering both structural and functional vulnerability. We computed interdependent effect between these networks under different conditions.

Most of the above discussed approaches has the deficiency of identifying vulnerable patterns, and protecting user information from malicious threats.

3. PROPOSED METHOD

we propose a naval sensitive operational attribute pattern based vulnerability analysis, using rule sets. The proposed approach has the following steps; Preprocessing, Pattern generation, Vulnerability analysis.


SOAP: Based Vulnerability Analysis
Pre Processing
Pattern Generation
Vulnerability Score Computation
Transactional Web log

3.1 Preprocessing:

The preprocessing is performed to remove noisy logs from the web log. At the preprocessing stage the web log is cleaned by identifying incomplete and noisy logs and converted into the form of computational script. The every access they get in which important piece of information about the accessing the recorded from the URL requested the IP address which the request invents and timestamp, service accessed and status of the service and etc…

Algorithm:

Step1: start

Step2: read web log Wl.

Step3: for each log l from wl

Check for the feature and noise.

If incomplete then

Remove record from log.

```
$Wl = Ø (l×Wl)$.
    End
Step4: stop.
```

3.2 Pattern Generation:

The sensitive operational attribute pattern is generated using the web log and for each log from the trace we identify the operations it followed or services invoked and the set of attributes being accessed. From the attribute set, we identify sensitive and non sensitive attributes and generate the SOAP pattern. The soap pattern represents how deeply service request has accessed the internal business logic and business data. So that, we can identify whether the pattern is malicious or not.

```
Input : Web log Wl.
Output: SOAP pattern Sp.
Step1: initialize soap pattern Sp, Initialize Attribute Lists Al.
Step2: for each log $Wl_i$ from Wl
Identify set of services followed $sf = \Sigma$ Services ( $Wl \times Wl_i$).
for each service $sf_i$ of sf
Identify set of attributes accessed $AA = \Sigma$ SA( $Sf_i \times Sf$)
compute Number of sensitive and Non sensitive attribute.
NSA = number of sensitive attributes from Al.
$NSA = Ø( AA \in Al)$.
    NNSA = number of non sensitive attributes.
$NNSA = Ø( AA \in Al)$.
end.
$Sp_i = \{\{Sname_i ,NSA_i ,NNSA_i \}, …., \{Sname_{i+n} ,NSA_{i+n} ,NNSA_{i+n} \}\}$.
    $Sp = \Sigma Sp+Sp_i$.
    end.
Step3: Stop.
```

3.3 Vulnerability Analysis:

At this phase, we compute vulnerability score of each pattern using the rule sets and computed Soap patterns. The vulnerability score is computed based on the completeness status of the service and number of sensitive and non sensitive attributes the service flow has accessed. We generate set of vulnerable patterns, so that future patterns can be identified easily.

```
Input: Soap Pattern Sp, Rule set Rs.
Output: Vulnerable Patterns vp.
Step1: for each soap pattern $sp_i$ of Sp
    for each rule R from Rs
compare whether the rule is match with the pattern.
    if $sp_i$ equals R
    Add to vulnerable pattern Vp.
    $Vp = \Sigma Vp +Sp_i$.
    end.
    end.
    end.
Step2: for each soap pattern $sp_i$ of Sp
    for each rule R from Rs
compare whether the rule is match with the pattern.
    if $sp_i$ equals R
        end.
    end.
    end.
```

```
        if No rule match then
If the service status is Failed then.
        count number of sensitive services NSS = $\Sigma$ Services ($Sp_i$).
        count number of non sensitive services NNSS = $\Sigma$ Services ($Sp_i$).
        Total number of sensitive attributes of Spi, Tsa = $\Sigma$ NSA( $Sp_i$).
        Total  non sensitive attributes of Spi, Tnsa = $\Sigma$ NNSA( $Sp_i$).
        If the last service invoked is Sensitive then
Vulnerability Score Vs =( (NSS×Tsa×0.8)+(NNSS×Tnsa×0.2))/(Tsa+Tnsa)
        else
Vulnerability Score Vs =( (NSS×Tsa×0.6)+(NNSS×Tnsa×0.2))/(Tsa+Tnsa)
        end.
        if Vs>0.6 then
        Add $Sp_i$ to Vulnerability pattern vp.
                Vp = $\Sigma$Vp+$Sp_i$.
        end
        end.
Step3: Stop.
```

4. RESULTS AND DISCUSSION

The proposed SOAP based vulnerability analysis framework has produced better results in identifying the malicious threats and network threats which happens in different business environment also in other transactional fields.

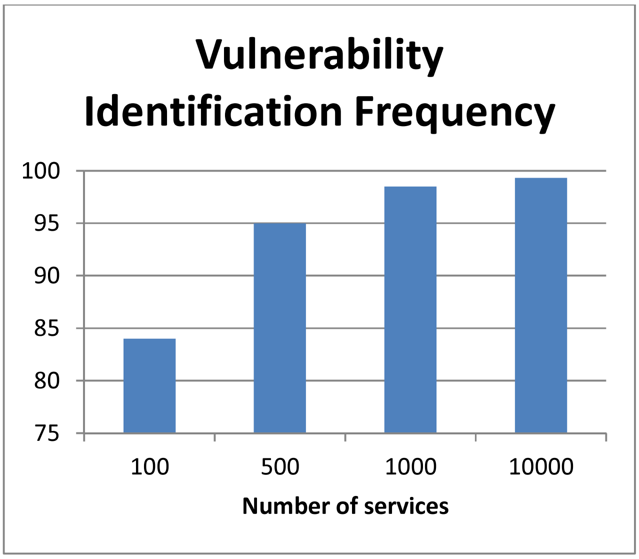
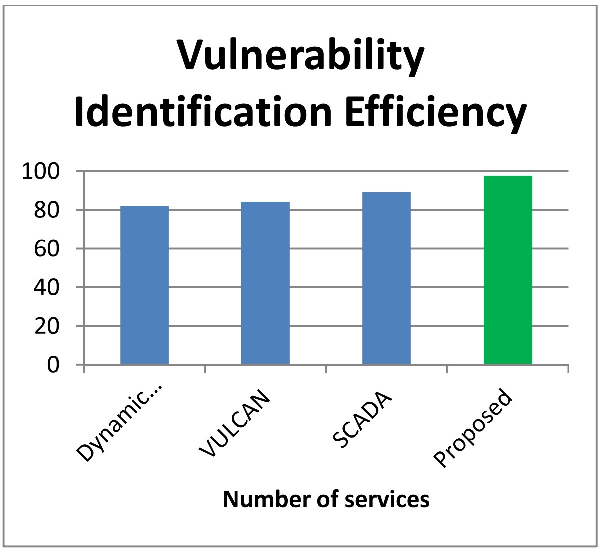


Table1: Shows The Frequency Of Identifying Vulnerability.

The table 1, shows the vulnerability identification frequency, it shows the frequency raises with the set of pattern, i.e. it is proportional to the number of patterns.



Graph2: Comparison Of Efficiency Of Different Methods.

The graph2, shows the efficiency of vulnerability identification, and it shows that the proposed approach has produced more efficiency in identification process.

5. CONCLUSION

We proposed a vulnerability analysis framework which uses the web log traces to preprocess and extracted the necessary features to generate sensitive operational attribute pattern SOAP using which we have computed the transactional completeness and vulnerability score to identify the vulnerability and propose set of patterns how the malicious user generates vulnerabilities to the business solutions.

REFERENCES

[1]. Unclassified Statement for the Record on the World wide Threat Assessment of the US Intelligence Community for the Senate Select Committee on Intelligence James R. Clapper Director of National Intelligence January 31, 2012.

[2]. “Towards a new stage in the bi-regional partnership: innovation and technology for sustainable development and social inclusion” MADRID ACTION PLAN 2010-2012.

[3]. Web Security Log Server Troubleshooting Guide Topic 50300 | Web sense Web Security Solutions | Version 7.7 | Updated 29-Jun-2012.

[4]. Decision-making under uncertainty: an assessment of adaptation strategies and scenario development for resource managers a white paper from the California energy commission’s California climate change center July 2012 cec-500-2012-027.

[5]. D.wivedi A, A Maximum-Flow-Based Complex Network Approach for Power System Vulnerability Analysis, Ieee Transaction on industrial informatics volume 9, issue 1, pp 81-88, 2013.

[6].Hossain Shahriar, Mitigation of Program Security Vulnerabilities: Approaches and Challenges, ACM, 2014.

[7].Yong Wang, Research of Network Vulnerability Analysis Based on Attack Capability Transfer, IEEE international conference on computer and information technology, pp:38-44, 2012.

[8].Shiguo Sun ,The Research on Network Vulnerability Analysis Methods [8], International conference on intelligent system design and engineering application, pp:593-597, 2012.

[9].Kamongi P, VULCAN: Vulnerability Assessment Framework for Cloud Computing, International conference on software security and reliability, pp:218-226, 2013.

[10].Hu Ruo, Information Security Vulnerability Analysis System Based on Dynamic cooperation Mechanism, WRI world congress on software engineering, vol.4, pp:142-149, 2009.

[11].Middleton.R, Global Network Security: A Vulnerability Assessment of Seven Popular Outsourcing Countries, International conference on green computing and communication, pp:102-108, 2012.

[12].Ju An Wang, Ranking Attacks Based on Vulnerability Analysis,Hawaii, International conference on System Sciences, pp:1-10, 2010.

[13]. Jun Xu, Software vulnerability analysis framework based on uniform intermediate representation, International conference on software technology and engineering, vol.1, pp:356,361,2010.

[14].Chopade, Structural and functional vulnerability analysis for survivability of Smart Grid and SCADA network under severe emergencies and WMD attacks, International conference on technologies for homeland security, pp:99-105, 2013.