IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Detecting Intrusions in Multitier Web Application

Shraddha Dabholkar[1], Rohit Khambe[2], Prof. Pallavi Chandratre[3]

Department of Computer Engineering, Mumbai University, Maharashtra, India[1,2,3]

Abstract: Internet services and applications have become an inseparable part of daily life, enabling communication and the management of personal information from any places. To contain this increase in application and data complexity, web services have moved to a multitier model how the web server runs the application front-end logic and data are outsourced to a database or file server. In this paper we are presenting double guard and IDS models the network behavior of user sessions across both the front-end web server and the back-end database. By observing both web and subsequent database requests, we are able to find out attacks that are not dependent IDS would not be able to identify. Furthermore, we compute the limitations of any multitier architecture in terms of working sessions and functionality coverage. We implemented Intrusion Detection system using an Glassfish web server with SQL Server 2014 and lightweight virtualization. We then collected and processed real-world traffic over a 15 day period of system deployment in both dynamic and static web applications. Finally, using Double Guard, we could expose a large range of attacks with 90 percent accuracy while maintaining 5 percent false positives for static web services and 5 percent false positives for dynamic web services.

Keyword: Multitier, Double Guard, Escalation attack.

I. INTRODUCTION

Web based attacks have recently become more various, as attention has shifted from attacking the front-end to utilizing vulnerabilities of the web applications in order to tainted the back-end database system (e.g., SQL injection attacks). A plethora of Intrusion Detection Systems (IDS) instantly examine overall network packets individually within both the web server and the database system. However, there is very little task being performed on multi tiered Anomaly Detection (AD) systems that produce models of network behavior for both web and database network interactions. In such multi tiered architectures, the back-end database server is safeguarded behind a firewall while the web servers are remotely accessible over the Internet.

Though they are safeguard from direct remote attacks, the back-end systems are susceptible to attacks that use web requests as a means to misuse the back-end. To protect multi-tiered web services, Intrusion detection systems (IDS) have been extensively used to detect known attacks by matching misused traffic patterns or signatures.

A class of IDS that power machine learning can also find out not known attacks by identifying abnormal network traffic that digress from the so called normal behavior that is previously profiled during the 3IDS training phase. Individually, the web IDS and the database IDS can find out uncommon network traffic sent to either of them. However found that these IDS can’t find out cases where as normal traffic is used to attack the web server and the database server.

II. EXISTING SYSTEM

Intrusion detection system currently inspects network packets individually within both the web server and the database system. But there is very less work being performed on multi-tiered Anomaly Detection system that creates architectures of network behavior for both web and database intercommunication.

In such multitier architectures ,the back end database server is usually protected behind a firewall where the web servers are remotely accessible over the internet. So that they are safeguarded from direct remote attacks, the back-end systems are manageable to attacks that use web request to misuse the back end

III. PROPOSED SYSTEM

In Double guard detection using both front end and back end detection. Some above perspective have detected intrusions or endangered by stable analyzing the source code. Other forcible track the information flow to understand infect propagations and detect intrusions. In double guard, the new holder based web server architecture enables us to separate the different data flows by each sessions by using light weight virtualization. Within a weightless virtualization territory we ran many copies of web server occurrences in different containers so that each one isolated from the rest

IV. ARCHITECTURAL DESIGN

Normal three tier Architecture

In fig it shows that the system contains three clients, web server and database server. Users send request to the web server, then web server take that request and send to database query to database server.

When web server sends database query to database at that time database take that query and process it and send back to the web server. Web server receives that database reply and send it to the client.

DOI 10.17148/IJARCCE.2016.5307

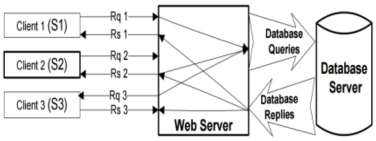

Fig.1 Normal three tier architecture

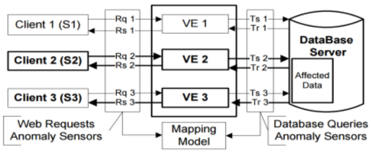

Fig.2Building Normality

V. ATTACK SCENARIO

A. Privilege Escalation Attack

In fig it shows privilege escalation attack, in this attack attacker use users information and try to take information from the database.

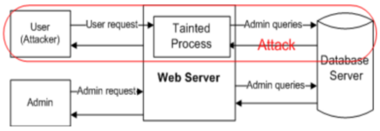

Fig.3 Privilege Escalation Attack

B. Hijack Session Attack

In this attack, when the user send the request to the web server at that time attacker attack on the web server and give them bogus reply the query is dropped it not get to the database.

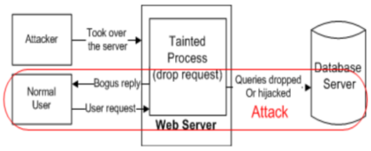

Fig.4 Hijack Session Attack

C. SQL Injection Attack

In this attack the attacker send injected user request to web server and inject the database queries and database send the replies to web server and web server give the privileged information to the attacker.

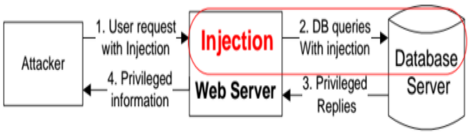
Attacker
1. User request with Injection
4. Privileged information
Injection
Web Server
2. DB queries With injection
3. Privileged Replies
Database Server

Fig.5 SQL Injection Attack

D. Direct Database Attack

It is possible for an attacker to neglect the web server or firewalls and connect directly to the database. An attacker is already there on the web server and be submitting such queries from the web server without sending web requests. Furthermore, if these Database queries were the set of allowed queries, then the database IDS itself would not detect it. However, this type of attack can be tackle with our approach since we cannot match any web to directly query the database.

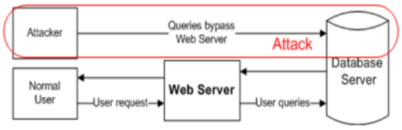

Fig.6 Direct Database Attack

VI. EXPECTED RESULTS

Our project aims at prevention of basically 4 types of attacksHijack Session attack- the attacker would not be able to send any bogus reply due to the kind of security we implemented. No outsider can easily check into the session. Privilege Escalation Attack- All the data in the database will remain safe and attacker will not be able to take any kind of information from DB. SQL Injection Attack- The Server will be able to detect if any malicious code in injected in the query and will try to prevent it accordingly, thus avoiding revelation of any sensitive data. Direct Database Attack- Our project does not match any web request directly to the database thus it makes impossible for the attacker to bypass and reach the database.

VII. CONCLUSION

Double guard detection presented an flaws detection system that builds models of normal behavior for multitier web applications to both front end web requests and back-end database queries. Unlike previous perspective that correspond or encapsulate alerts generated by independent IDS, Double Guard forms a container-based IDS with n number of input streams to create alerts. We have shown that such correlation of input streams gives a better characterization of the system for abnormality detection because the intrusion sensor has a more accurate normality model that detects a extensible range of threats. This system achieved this by separating the flow of information from each web server session with a weightless virtualization .so that, we quantified the detection accuracy of our approach when we crack to static model and dynamic model with the back end file system and database queries. For stable websites, we made a well-correspond architecture our experiments proved to be effective at detecting different types of attacks.

REFERENCES

[1]. V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, “Toward Automated Detection of Logic Vulnerabilities in Web Applications,” Proc. USENIX Security ymp., 2010.

[2]. G. Vigna, W.K. Robertson, V. Kher, and R.A. Kemmerer, “A Stateful Intrusion Detection System for World-Wide Web Servers,” Proc. Ann. Computer Security Applications Conf. (ACSAC ’03), Oct.2003.

[3]. C. Kruegel and G. Vigna, “Anomaly Detection of Web-Based Attacks,”Proc. 10th ACM Conf. Computer and Comm. Security (CCS ’03), Oct. 2003.

[4]. Liang and Sekar, “Fast and Automated Generation of Attack Signatures: A Basis for Building Self-Protecting Servers,” SIGSAC: Proc. 12th ACM Conf. Computer and Comm. Security, 2005.

[5]. Y. Hu and B. Panda, “A Data Mining Approach for Database Intrusion Detection,” Proc. ACM Symp. Applied Computing (SAC), H. Haddad, A. Omicini, R.L. Wainwright, and L.M. Liebrock, eds., 2004.