IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com

ISSN 2348 – 7968

Intrusions Detection System with Double Armor in Multitier Web Applications

Prof. R. B. Rathod, Jayant Gaikwad, Jayant Gaikwad, Swapnil Kotwal, Prashant Kalel

PDEA's College of Engg. Manjari(Bk).Pune

Abstract— **Now a days there is much more information has to be managed from anywhere in the World. So this increases data complexities. To control this type of problem all data can be moved from single tier to multitier web applications that means front end and back end. And just like that there is also increase of intrusions in this scenario. Intrusion detection systems are used to detect attacks which can harmful to the computer and network. In this Paper we present IDS to detect and identify intrusions in both front end web servers and back end database servers. Traditional IDS can only used to detect attacks only at single side. We also quantify the drawbacks and limitations of multitier IDS in terms of delay, throughput, speed, tracking sessions, false negative ratio etc. Apache web server with MYSQL can be used for implementing multitier IDS in this paper. We have analyzed various types of traditional IDS for this purpose and their limitations can be eliminated .Finally, using Double Armor, we are able to manage accuracy, and less delay in web services.**

Index Terms— **Multitier, intrusions, Intrusion detection system (IDS), Apache web server, MYSQL**.

I. Introduction

Internet services become essential feature of daily life. So it is fitted into daily life for increase in data complexity, application. Web services moved to multitier to provide high reliability in accessing those services for all users. So it becomes very easy for the customers to interact with it. But by creating multitier for information storage is very easy for attacker to attack on to the both the tiers. So there is need to provide security to both the tiers. Double armor is an Intrusion Detection system in which we can

track the sessions of front end web server as well as back end databases. By identifying both the front end web server and back end database server we can detect various types of attacks which single independent IDS cannot used to detect the attacks. We can ferret out any limitations of multitier IDS in terms of delay, accuracy. Double armor is implemented using an APACHE server and MYSQL

An Intrusion Detection System (IDS) can currently detect or examine network packet individually within both database server and web server. To protect the multitier web applications, Intrusion Detection System (IDS) is widely used to detect known attacks from matching it with various patterns and signatures. It also can be used to detect unknown attack by identifying abnormal network traffic which deviates from Normal activities which can be tabled into the IDS. Individually, the database IDS and Web IDS can detect only those attacks which are coming to them only. However these types of IDS will not work if the case that normal behavior is used to attack on database server and Web server. For example, if a non-admin privilege user can accesses the web server with a normal user credentials. He/she can find the ways to add vulnerabilities into the web server. This type of attack can neither be detected from database IDS nor the Web IDS since database IDS only see the network traffic of privilege user and web IDS can be used to see only user login traffic To detect this type of attack the database IDS can identify privilege request from the web server with non-privilege access. Such detection is very helpful in network traffic and it is used to prevent any unwanted misbehavior.

II. INTRUSION DETECTION SYSTEM

A. Definition

An intrusion detection system (IDS) is a mechanism to detect abnormal or suspicious activity on a given target to address the problems as quickly as possible. Given their practical value, the IDS have been studied heavily over the past 20 years in order to improve their effectiveness. The fruits of these studies are of different classes of IDSs that rely on different detection techniques, each of which is more appropriate for a particular context. Among others, we find the intrusion detection systems that base their decisions on information found in machines called HIDS and intrusion detection systems that base their decisions solely on information flowing in a network called NIDS.

B. VULNERABILITY OF SYSTEMS

An attack is an exploitation of vulnerability in a system. Thus, reducing attacks can only be done with a good understanding of the system and possible sources of vulnerability in order to find suitable remedies. The word vulnerability expresses all the weaknesses of computer resources that can be exploited by malicious people. D. Denning explains the presence of vulnerabilities in information systems by, among others, the following reasons:

 Good security costs usually very expensive and most organizations do not have sufficient budget to afford this need.

 Security tools used cannot be 100% sure, see that they are often ineffective.

 Security policies are commonly complex, incomplete and sometimes inconsistent.

 The bugs in programs that are common and are still exploited by attackers.

C. INTRUSION DETECTION SYSTEM : ISSUES

The Intrusion Detection System (IDS) has become a critical component of wireless sensor networks security strategy. However, deployment of intrusion detection brings with it a number of potential pitfalls, which can compromise security. Some of the issues related to ids in sensors network are:

A. It is not possible to have an active full-powered agent inside every node in a sensor network. Each node is totally independent, sending data and receiving control packets from a central system called Base Station, usually managed by a human user.

B. An IDS for sensor networks must send the alerts to the base station in order to warn the human user.

C. An IDS must be simple and highly specialized for reacting against specific sensor network threats and to the specific protocol used over the network.

D. ***Heavy traffic networks***

- In these, the high amount traffic overloads IDS sensor & intrusion traffic is missed.

E. ***Switched networks.***

- In these, an IDS needs to see the traffic on each switch segment. In these switched networks there is no location to connect n IDS – & switch SPAN ports can't keep up with all the traffic on the switch. Deploying IDS on each segment is cost prohibitive in many environments, thereby leaving segments unprotected.

F. Asymmetrical networks.

- In asymmetrically routed networks the traffic can traverse multiple paths before it reaches the n IDS and the n IDS will only see parts of the conversation (flow); thus missing an attack. An IDS needs to see a complete conversation (flow) in order to determine if an intrusion is present.

III. SYSTEM ARCHITECTURE

The classic three-tier model. At database side, we are not able to tell which transaction corresponds to which client request. The communication between Web Server and the database server isn't separated, and we can understand the relationships between clients and server. If Client 2 is malicious and takes over Web Server, all subsequent database transactions become suspect, as well as the response to client. By contrast, according to Fig. 1, Client 2 will only compromise VE 2, and the corresponding database transaction set T2 will be the only affected section of data within database.

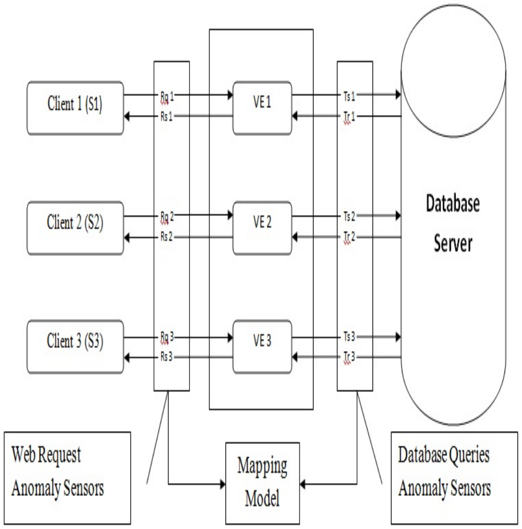

Fig. 1 System Architecture

This container-based and session-separated web servers architecture are not only enhances the security performance but also provides us with an isolated information flows that are divided in each container session. It allows us to detect the mapping between web server request and subsequent DB queries, and to utilize such a mapping model to identify abnormal behaviors on a session/client level. In typical three-tiered Web Server Architecture, Web Server receive HTTP requests from clients and then issues SQL queries to the database Server to fetch and update data. These SQL queries are dependent on the web request hitting the Web Server. We want to model such a causal mapping relationships of all legitimate traffic so as to identify abnormal/attack traffic.

Attack Scenarios:-

❖ *Privilege Escalation Attack*

This type of attack show how a normal user may use admin queries to obtain such a privileged-information. The attackers log into the web server as a normal user, upgrades their privileges, and triggers admin queries so as to get the administrator's information. Privilege Escalation attack can never be identified by either Web Server IDS or the database IDS

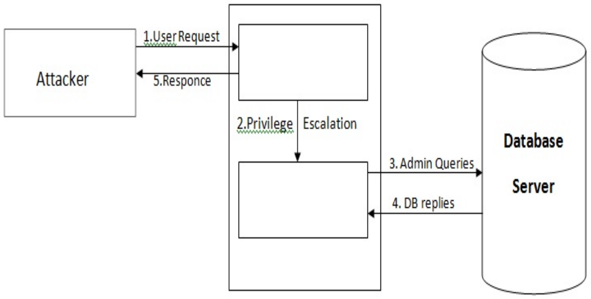

Fig.2. Privilege Escalation Attack

❖ ***Hijack Future Session Attack***

These attacks show the scenario where in a compromised web server can damage all the Hijacked Future Sessions by not creating any DB queries for the normal user requests. This attack is mainly focused at Web Server-side. An attacker usually takes over the web server and therefore hijacks all subsequent authorized user sessions to launch attacks. For instance, by hijacking other user's session, the attacker can eavesdrop, send spoofed replies, and/or delete user requests

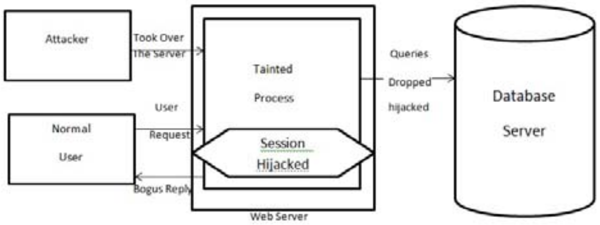

Fig.3. Hijack Future Session Attack

❖ *Injection Attack*

Attacks such as SQL injection don't need compromising Web Server. Attackers can use existing vulnerabilities in the web server logic to inject the existing data or string content that contains the exploits and then use the web server to relay these exploits to attack the database. An attacker could also have already taken over the web server and be submitting such queries from the web server without sending web requests.

Without matching web requests for such queries, a web server-side IDS could identify neither. Furthermore, if these database queries were within the set of allowed queries, then the database IDS it-self would not identify it either. However, this type of attack can be caught with Double Armor approach.

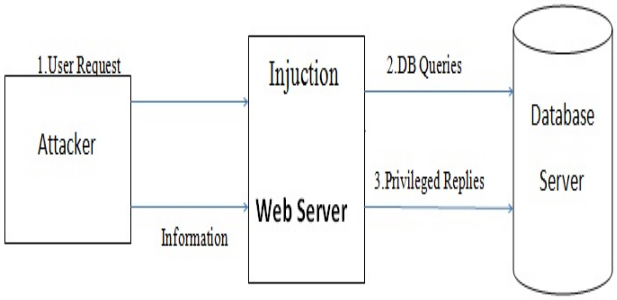

Fig.4. Injection Attack

IV. ADVANTAGES

1. Verification of attack's success or failure: Since the host based Intrusion Detection System uses system-logs containing events that have actually occurred, they can check whether an attack could be found or not with the greater accuracy and fewer false positive ratio than a n/w based system. Network based IDS sensors although faster in response than host based IDS sensors, generate a lot of false positives because of the very fact that it identifies malicious packets on the real time and some of these packets could be from the authorized host.

2. System Activities monitoring: An IDS sensor tests user and file access activity including file accesses, changes made to file permissions, attempts to install the new executables etc. An IDS sensor can also test all user log-on and log-off activities, user activities while connected to the n/w, file system changes made, activities that are normally executed only by an admin. OS log any event where user accounts are added, dropped or upgraded. The IDS can identify an improper changes made as soon as it is executed. A network-based system cannot give the detailed information about system activities.

3. Detection of network-based attacks: Network based IDS sensors can detect attacks, which host-based sensors fail to detect. A network based IDS checks for all the packet headers for any malicious attack. Much IP-based denial of service attacks like TCP SYN attack, fragmented packet attack etc. can be detected only by looking at the packet headers as they travel across a network. A network based IDS sensor can quickly identify this type of attack by looking at the contents of the packets at the real time.

4. Retaining evidence: Network based IDS use live network traffic and does real time Intrusion Detection. Therefore, the attacker cannot delete evidence of attack. This data can be used for forensic analysis. On the other hand, a host-based sensor identifies attacks by looking at the system-log files. Lot of hackers are capable of making changes in the log files so as to delete any evidence of an attack.

5. Detection at real time and quick response: Network based IDS tests traffic on a real time. So, network based IDS can detect malicious activity as they are occurring. Based on how the sensor is configured, such attack can be stopped even before they can get to a host and compromise the system. On the other hand, host based systems identify attacks by looking at changes made to the system files. By this time critical systems may have already been compromised.

V. CONCLUSION

We propose an intrusion detection system which builds models of normal behavior for multi-tiered web applications from both front-end web (HTTP) requests and back-end database (SQL) queries. Previous IDSs correlated or summarized alerts, whereas Double-Guard forms container-based IDS with multiple input streams to produce alerts. Such correlation of input streams provides a better characterization of the system for anomaly detection since the intrusion sensor has a more specific normality model that investigates a wider range of attacks.

REFERENCES

[1] K. Bai, H. Wang, and P. Liu. Towards database firewalls. In *DBSec 2005*.

[2] B. I. A. Barry and H. A. Chan. Syntax, and semantics-based signature database for hybrid intrusion detection systems. *Security and Communication Networks*, 2(6), 2009.

[3] D. Bates, A. Barth, and C. Jackson. Regular expressions considered harmful in client-side xss filters. In *Proceedings of the 19th international conference on World Wide Web*, 2010.

[4] M. Christodorescu and S. Jha. Static analysis of executables to detect malicious patterns.

[5] M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna. Swaddler: An Approach for the Anomaly-based Detection of State Violations in Web Applications. In *RAID 2007*.

[6] H. Debar, M. Dacier, and A. Wespi. Towards a taxonomy of intrusiondetection systems. *Computer Networks*, 31(8), 1999.

[7] V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna. Toward Automated Detection of Logic Vulnerabilities in Web Applications. In *Proceedings of the USENIX Security Symposium*, 2010.

[8] Y. Hu and B. Panda. A data mining approach for database intrusion detection. In H. Haddad, A. Omicini, R. L. Wainwright, and L. M. Liebrock, editors, *SAC*. ACM, 2004.