Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Survey on Techniques for Efficiently Detecting Intrusions in Multi-tier web Applications

Yalanati Ayyappa,
Student, Department of CSE,
VR Siddhartha Engg. College
India

K.Praveen Kumar
Sr.Asst. Professors, Dept. of CSE,
VR Siddhartha Engg. College
India

Dr. V.Srinivas Rao
Professor & Head, Dept. of CSE,
VR Siddhartha Engg. College
India

Abstract: ***Web-delivered service is an emerging approach for IT service to reduce cost and improve delivery efficiency. Web-delivered services and applications have increased in both popularity and complexity. Daily tasks, such as banking, travel, and social networking, are all done via the web. Now the web applications have moved to the multi-tier architecture. In multi-tier architecture the business logic, data access and data storage are separated because it has a very good advantage. If any modification has to be done no need to modify the entire application, instead the modification is done to the specific tier where it is required. In recent times web applications suffers different types of vulnerabilities. They are SQL injection attack, Privilege Escalation Attack, session hijacking attack, direct DB attack, Denial of Service (DOS) attack and cross site scripting attack. We studied several existing systems to defend against these attacks.***

Keywords: vulnerabilities, SQL injection, session hijacking, denial of service attacks, intrusion detection system, multi-tiered web application

1. INTRODUCTION

An *intrusion detection* system is a device or software application that monitors network and system activities for malicious activities or policy violations and produces report to the management station. *Multi-tier architecture* is a logical partition of the separation of operations in the system. Each tier is assigned its unique responsibility in the system. Multi-tier architectures typically comprise a presentation tier, a business or data access tier, and a data tier which are logically separated. A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface.

Web pages can be either static or dynamic. "Static" means unchanged or constant. Static Web pages contain the same content each time the page is loaded. While "dynamic" means changing. Virtualization allows a single computer to do the job of many computers. Virtualization technology can save money and simplify IT operations. A light weight virtualization technical is used in the intrusion detection system. It is a container based approach which holds/manages the user sessions in containers. Advantage is thousands of containers can run on a single host. An attack is any attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset.

Attacks are several types these are

1. Privilege Escalation Attacks
2. Hijack future session attacks
3. SQL injection attacks
4. Direct DB attack
5. Denial of service

1.1 Privilege Escalation Attacks: Privileges means what a user is permitted to do. Privilege escalation means a user receives privileges they are not entitled to. These privileges can be used to delete files, view private information (or) Install unwanted programs such as virus.

Privilege escalation occurs in two forms

1. Vertical privilege escalation.
2. Horizontal privilege escalation.

Vertical privilege escalation means where a lower privilege user can access site administrative functions.

Horizontal privilege escalation occurs when an application allows the attacker to gain access to resources which normally protected from an application or user.

1.2 Hijack future session attacks: Session hijacking is the misuse of a valid computer session, sometimes also called a session key to gain unauthorized access to information (or) services in a computer.

There are four main methods used to commit a session hijack.

1. Session fixation.
2. Session side jacking.
3. Attempt to steal the session key.
4. Cross-site scripting

1.3 SQL injection attacks: Databases are fundamental components of Web applications. Databases enable Web applications to store data, preferences and content elements. Using SQL, Web applications interact with databases to dynamically build customized data views for each user.

SQL injection is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a Web application for execution by a backend database. Attackers take advantage of the fact that programmers often chain together SQL commands with user-provided parameters, and can therefore embed SQL commands inside these parameters. The result is that the attacker can execute arbitrary SQL queries and/or commands on the backend database server through the Web application.

1.4 Direct DB attack: An attacker to bypass the web server or firewalls and who is directly connected to the database can submit the quires without sending web requests from the web server. Such attacks cannot be detected by web server intrusion detection system and database intrusion detection system.

1.5 Denial of service attack: A denial-of-service attack (DoS attack) is an attempt to make a machine or network resource unavailable to its intended users.

2. Related Work

Meixing Le el.al., It detects four types of attacks i.e. sql injection attack, privilege escalation attack , hijack future session attack, direct DB attack. Using a lightweight virtualization technique to assign each user web session to a dedicated container. Each user traffic is separated to other users by using containers. If a session is attack then that session will be effected other sessions should not effected.

In this method mapping is provided between web server and database server for checking the session id in webserver and database server if the both session id's will be same then it allow to access the database, otherwise there is a vulnerability .

In this method we identify the some drawbacks, it can't detect cross side scripting attacks and distributed DOS attacks.

Romaric Ludinard et.al., Romaric Ludinard proposed a ruby on rails anomaly based intrusion detection system. It is an application level intrusion detection system for applications implemented with ruby on rails frame work.

In this approach we form invariants on variables. By checking the invariants on run time to detect the attacks. In this approach we are using ruby code to authenticate the user. User login and password are passed as parameters and stored in the hash table variable. A method user.authenticate is used to retrieve the actual user to corresponding to these parameters. In this attribute login is equal to the login passed parameters is an invariant. Three types of invariants are used in this approach

- Relations between variables at a given execution point.
- Relations between the different values held by a given variable at different points of execution.
- Relations between different variables at different points of execution.

In this execution point plays important role. The execution point is a triplet. It consists of a

- Program counter
- The state of the memory associated with the program
- Logical time

We find invariants that are valid for all execution points.

In this Approach SQL injection attacks are detected very efficiently.

The disadvantage of this approach is it cannot detect the Cross site scripting attacks and Session hijacking attacks.

Dessiantnikoff et.al., To identify the vulnerabilities in the web application we are used Vulnerability scanners .The scanners are analyzed the web applications and it detects the vulnerabilities in the web application. The disadvantage of scanners is may exhibit a significant large number of false positives and false negatives.

Dessiantnikoff proposed an algorithm that uses the clustering technique. In this approach injection point plays a major role. Dessiantnikoff approach is depends the following assumptions

- The content of an execution page is far differing from the content of a rejection page.
- Two rejection pages may different from each other.
- Two execution pages may also different from each other.

In this approach focus is on the analysis of differences. The main aim is to tell whether the response is a rejection page or execution page. A threshold is used to determine whether the two responses are similar or not. The threshold value is depending on the size of the responses and the amount of data that change between the two responses. In this approach we are efficiently detecting the SQL injection attacks. Other types of attacks are not detected.

Cristian Pinzon et.al., Cristian Pinzon proposed an case-based reasoning intrusion detection system for sql injection attacks. In this a new problem is solved by using the case memory to get a similar case which case been solved in the past. In this a CBR agent is used to study the internal structure and the classification mechanism of SQL attacks. This mechanism is a

combination of artificial neural network and a support vector machine in order to provide most reliable classifier to detect queries that are malignant.

The case-based reasoning consist of the following

- To generate the plan, the problem description is used.
- The problem statement consists of
  - Case identification
  - User session
  - SQL elements
- Prediction models are used to solve the problem.
- We achieve the final state after applying the solution.

In this approach different stages are their

- Retrieve
- Reuse
- Revise
- Retain

The retrieve phase is consists of two parts, case retrieve and model retrieval. In the case retrieve step the queries are retrieved from the case memory. Model retrieve step is used to improve the system performance. The reuse phase is used to reduce the false negatives. The revise phase can be done automatically or manually depending upon the output values. The output values from 0.35 to 0.6 then the case is suspicious, a human expert is performed a review on the case. The retain phase is used to update the information when a new case is entered. In this method only SQL injection attacks are detected. Other types of attacks can't be detected by the system.

Sekar et.al., In this method the system is placed in web server/web application architecture using library interposition in order to observe the incoming and outgoing requests. The events are collected by the event collector. The event inspectors include a pluggable architecture to do the syntax analysis. The information is in the form of <name, value> to simplify the design. The taint interface algorithm is operating on single data item set at a time. The output of the syntax analyzer is to construct the abstract syntax tree. If any policy violations then the output request are blocked by the event inspector and error code will be return to the caller. The advantage of the approach is no false negatives and no false positives. It detects only SQL injection attacks and cross-site scripting attacks with single framework. Using fewer policies it can detect attacks very efficiently. Session hijacking attacks and direct db attacks can't detect this system.

Table-1
COMPARISION OF DIFFERENT TECHNIQUES AND ATTACK DETECTIONS

Attacks / Techniques	Sql injection attacks	Privilege Escalation Attacks	Hijack future session attacks	Direct DB attacks	Denial of service attacks	cross site scripting attacks
Virtualization technique	Yes	Yes	Yes	Yes	Yes	No
Invariants based technique	Yes	No	No	No	No	No
Clustering technique	Yes	No	No	No	No	No
case-based reasoning technique	Yes	No	No	No	No	No

library interposition technique	Yes	No	No	No	No	Yes

4. CONCLUSION AND FUTURE WORK

In this study we identify that multi-tier web applications mostly suffer with Privilege Escalation Attacks, Hijack future session attacks, SQL injection attacks, Direct DB attacks, Denial of service attacks, cross site scripting attacks. But we do not identify a system to detect all these attacks by using a single system. In future work we are proposing to derive an efficient system to detect all these attacks**.**

REFERENCES:

[1]. Double Guard: Detecting Intrusions in Multi-tier web applications Dependable and Secure Computing, IEEE Transactions on JulyAug.2012.

[2]. A clustering approach for web vulnerabilities detection 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2011), Pasadena.

[3]. An Efficient Black-box Technique for Defeating Web Application Attacks IEEE 2010.

[4]. CBRID4SQL"A CBR intrusion detector for SQL injection attacks IEEE 2010.

[5]. Detecting attacks against data in the web application IEEE 2010.

[6]. http://wiki.openvz.org.

[7]. Linux-vserver, http://linux-vserver.org/, 2011.

[8]. Joomla cms, http://www.joomla.org/, 2011.

[9]. Autobench, http://www.xenoclast.org/autobench/, 2011.K. Bai, H. Wang, and P. Liu, "Towards Database Firewalls," Proc. Ann. IFIP WG 11.3 Working Conf. Data and Applications Security (DBSec '05), 2005.

[10]. B.I.A. Barry and H.A. Chan, "Syntax, and Semantics-Based Signature Database for Hybrid Intrusion Detection Systems," Security and Comm. Networks, vol. 2, no. 6, pp. 457-475, 2009.

[11]. M. Christodorescu and S. Jha, "Static Analysis of Executables to Detect Malicious Patterns," Proc. Conf. USENIX Security Symp.,2003.

[12]. H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion-Detection Systems," Computer Networks, vol. 31, no. 9, pp. 805-822, 1999.

[13]. V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, "Toward Automated Detection of Logic Vulnerabilities in Web Applications," Proc. USENIX Security Symp., 2010.

[14]. A.Seleznyov and S. Puuronen, "Anomaly Intrusion Detection Systems: Handling Temporal Relations between Events," Proc. Int'l Symp. Recent Advances in Intrusion Detection (RAID '99), 1999.

[15]. Srivastava, S. Sural, and A.K. Majumdar, "Database Intrusion Detection Using Weighted Sequence Mining," J. Computers, vol. 1,no. 4, pp. 8-17, 2006.

[16]. F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation,"IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.

[17]. 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), 2003.

[18]. G. Vigna, F. Valeur, D. Balzarotti, W.K. Robertson, C. Kruegel, and E. Kirda, "Reducing Errors in the Anomaly-Based Detection of Web-Based Attacks through the Combined Analysis of Web Requests and SQL Queries," J. Computer Security, vol. 17, no. 3,pp. 305-329, 2009.

[19]. P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kru¨ gel, and G. Vigna, "Cross Site Scripting Prevention with Dynamic Data Tainting and Static Analysis," Proc. Network and Distributed System Security Symp. (NDSS '07), 2007.

[20]. D. Wagner and D. Dean, "Intrusion Detection via Static Analysis," Proc. Symp. Security and Privacy (SSP '01), May 2001.

