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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Distributed Intrusion Detection System to Protect Enterprise Web Applications

Pravallika.P[1], Radha.R[2]

Student, Department of CSE, MRCET, Hyderabad, India [1]

Asst.Professor, Department of CSE, MRCET, Hyderabad, India [2]

Abstrac: Web applications have become common for organizations as they can allow access to information and operations without geographical and time restrictions. For improved service quality web applications are built in multi-tiered architecture. There are many tiers involved such as client tier, web tier and data tier. The browser comes under client tier. The web server running web resources come under web tier while the database comes under data tier. Most of the existing Intrusion Detection Systems (IDS) are capable of protecting either web server or database server. They can't provide end to end security that covers web server and database server. Recently Le et al. presented a secure mechanism that covers the entire architecture. In this paper we implement that mechanism by building a multi-tiered application using Servlets and JSP in web tier and MY SQL as backend. The experiments revealed that the proposed mechanism is capable of extending protection to the web server and also database server.

Index Terms: Distributed intrusion detection, multi-tier web application, anomaly detection

I. INTRODUCTION

Over the past many years web based applications gained in popularity. They are also complex due to the new innovative technologies and the use of Web 2.0. Every day operations are made through various web portals in domains like banking, insurance, health care, government and so on. These applications are typically in three tier architecture. The three tiers include client tier, web tier, and data tier. Enterprise applications use this architecture to provide services online. Attackers make use of the vulnerabilities present in three tier architecture. They attack on both web tier [1], [2], [3] and also database tier [4]. The database tier attacks are mostly SQL injection attacks [5], [6]. Currently there are number of intrusion detection systems (IDSs). However, they focus on either database tier or web tier but not both. Some IDSs also prevent attacks made by adversaries by matching misused signatures and traffic patterns [7], [8], and [9].

Recently Le et al. [10] proposed a novel IDS namely DoubleGaurd. This IDS covers both web tier and database tier. In fact it provides integrated security. In order to achieve this, the authors used light-weight visualization technique in a session based virtual environment. Container ID is used to map request to corresponding DB traffic. They implemented container architecture of DoubleGuard using OpenVZ [11]. The empirical results reveal that the IDS is effective.

In this paper we have implemented the architecture provided in [10] in a three tier architecture. We built a prototype application that demonstrates the proof of concept. The architecture uses web browser, web server such as Tomcat 7.0 and database server such as MY SQL. The technologies used for building front end include Servlets and JSP pages besides HTML and JavaScript. The application is protected by the DoubleGaurd. The DoubleGuard has abilities to prevent attacks made through web server and database server. The empirical results revealed that the proposed IDS system is very effective as it can protect the whole application at web server as well as database server. The remainder of the paper is structured as follows. Section II reviews literature. Section III provides details about proposed system. Section IV presents prototype implementation details. Section V presents experimental results while the section VI concludes the paper.

II. PRIOR WORKS

This section reviews literature on IDSs. The existing IDSs can be classified into two categories. They are known as misuse detection and anomaly detection. The later needs the IDS to define dynamic and static characteristics which can be used later to detect intrusions with ease [48], [12]. There models for statistical analysis and also rule-based approaches [13], [14], [15], and [16]. There are intrusion alert correlation systems [17] that are made up of a

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

collection of components that can transform the alerts of IDS into compact intrusion reports that contain reduced alerts. The duplicate alerts, false positives and no relevant positives are automatically detected and removed. Primarily it focuses on the low-level, logical and high-level alerts. However, the proposed system differs from this approach but correlates alerts coming from independent IDSs. Multiple feeds are used in IDS with multiple sessions.

Highest level of protection has to be made to databases. This is because very valuable information is stored in databases. The business data is an asset to every organization. For this reason significant research has been made on IDS that focus in this area [18], [19], and [20]. There was also research made on firewalls [21]. Green SQL [22] kind security products act as reverse proxy in order to secure database connections that come from web clients. With this in place, the web servers do not connect to database servers directly. Instead they connect to database firewall prior to going to database. To achieve more protection in [23] both database IDS and web IDS are used with the help of reverse HTTP proxy.

Some of the existing approaches used source code for vulnerability analysis. They include [24], [25], and [26]. Recently Le et al. [10] proposed IDS that covers both database server and web server security. Hence, the name “DoubleGuard”. This will track the requests right from browser to database server based on sessions. Application logic building model is not required by DoubleGuard. However, full application logic is not required for dynamic web services. SQL Injection attack and Cross Site Scripting (XSS) are the attacks that can be detected using input validations. However, DoubleGuard used input validations only as an additional defence mechanism. Virtualization technique also used to improve security performance. They used light weight virtualization technique like Linux – Vserver [27], Virtuozzo [28] and Open VZ [11]. All these can be used as container. Desktop systems can also use the concept of virtualization [29]. For preventing data leaks CLAMP [30] is used. It provides guaranteed security to the sessions of different users.

In contrast to CLAMP, DoubleGuard maps HTTP requests to DB queries and prevent malicious user sessions. There are additional defenses in DoubleGuard which are not in CLAM. DoubleGuard is effective to handle various kinds of attacks.

III. PROPOSED SYSTEM FOR PREVENTING ATTACKS

The proposed system assumes that both database server and web server are vulnerable to various attacks such as privilege escalation attack, hijack future session attack, and Injection attack. The three attack scenarios are described below.

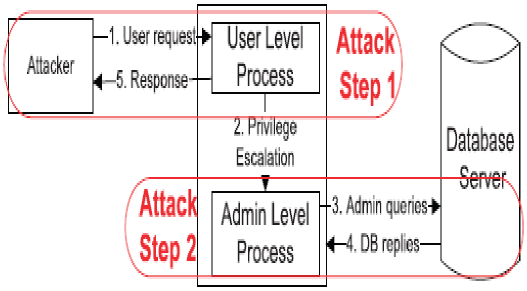


Fig. 1 – Illustrates privilege escalation attack (excerpt from [10])

As can be seen in fig. 1, the privilege escalation attack is illustrated. Through this attack the user who has privileges to user level process is able to escalate it to admin level process. When this attack is successful, the user can have administrator privileges and can misuse the system.

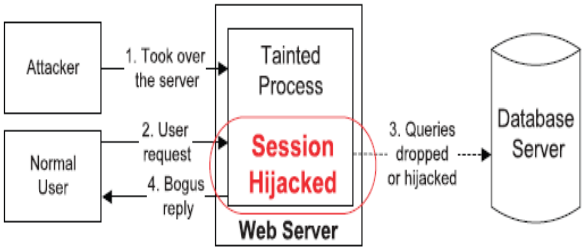


Fig. 2 – Illustrates Hijack Future Session Attack (excerpt from [10])

As seen in fig. 2, the hijacking of future session is illustrated. It takes place in many steps. In the first step an attacker takes control over the server. Then sends requests by hijacked session where the databse queires are eighter hijacked or dropped and gets bogus reply.

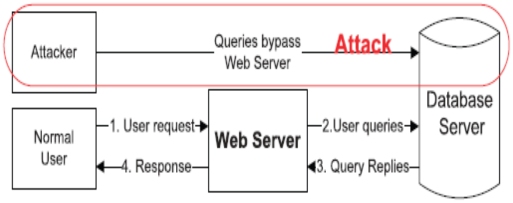


Fig. 3 – Illustrates Injection Attack (excerpt from [10])

As can be seen in fig. 3, the injection attack takes place when attacker bypasses web server and make direct DB requests. This kind of attack is very dangerous as the adversaries are able to gain access to DB directly.

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

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

Proposed Architecture to Prevent Attacks

In order to prevent all kinds of attacks and also protect database server and also web server, the following architecture is used. The architecture and corresponding algorithms are described in more detail in [10].

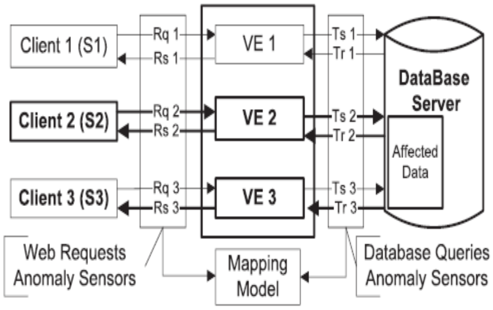
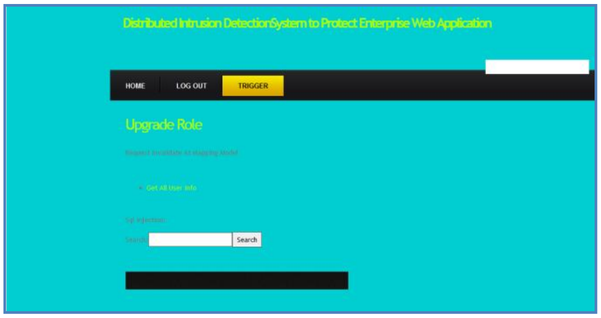


Fig. 4- Illustrates architecture that prevents attacks

As seen in fig. 4 web request anomaly sensors that help in mapping the requests to corresponding database queries. At database server also there are database queries anomaly sensors. These two kinds of sensors ensure that all types of attacks are prevented. This is because adversaries can't intrude either into database or web server due to the mapping and the presence of sensors at both layers. More information can be found in [10].

IV. PROTOTYPE IMPLEMENTATION

The proposed security mechanisms are implemented in such as way that they can prevent various kinds of attacks covering all tiers of the web application. The security mechanisms are capable of protecting the web applications from attacks such as privilege escalation attack, hijack future session attack, injection attack and direct DB attack. To demonstrate attacks we built front end UI which is presented here.


Distributed Intrusion DetectionSystem to Protect Enterprise Web Application
HOME
LOG OUT
TRIGGER

Upgrade Role

Request Invalidate At Mapping Model

- Get All User Info

Sql Injection:

Search: Search

Fig. 2- Privilege   Escalation Attack

As can be seen in fig. 2, the user interface facilitates an attacker to perform privilege escalation attack which tries to alter the privileges of a user to have higher credentials to gain access to sensitive areas of the database and web server. The user interface in fig. 2 also supports making SQL injection attack also.

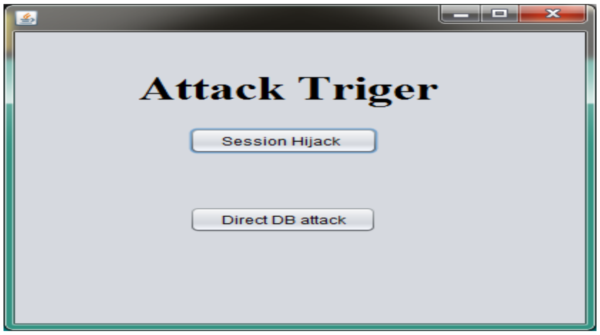


Fig. 2 - Hijack Future Session Attack

As can be seen in fig. 2, the interface allows user to make two kinds of attacks. They are future session hijack attack and direct DB attack. The future session hijack attack tries to hijack user sessions for monetary or other benefits. The direct database attack bypasses the request from going to regular path through web server to database server directly.

V. EXPERIMENTAL RESULTS

We have made many experiments with proposed security mechanisms. The parameters considered for experiments include HTTP load, reply rate, startup time of new container, training time versus false positives and so on.

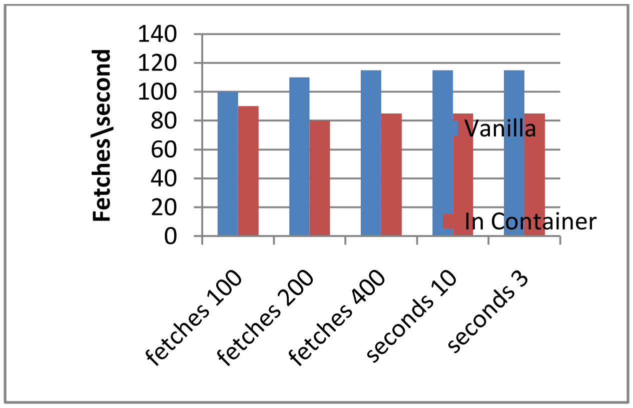


Fig 5. Performance evaluation using http load. The overhead is between 10.3 to 26.2 percent.

As shown in fig.5 the horizontal axis is represented as fetches while the vertical axis represented as fetches per second.

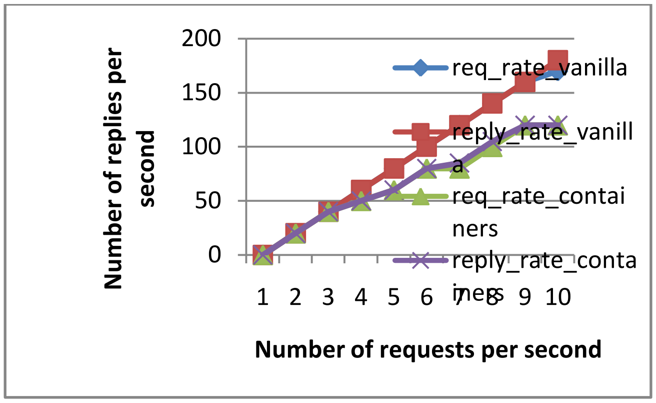


Fig 6. Performance evaluation using autobench.

As shown in fig.6 the horizontal axis is represented as number of requests per second while the vertical axis represented as number of replies per second.

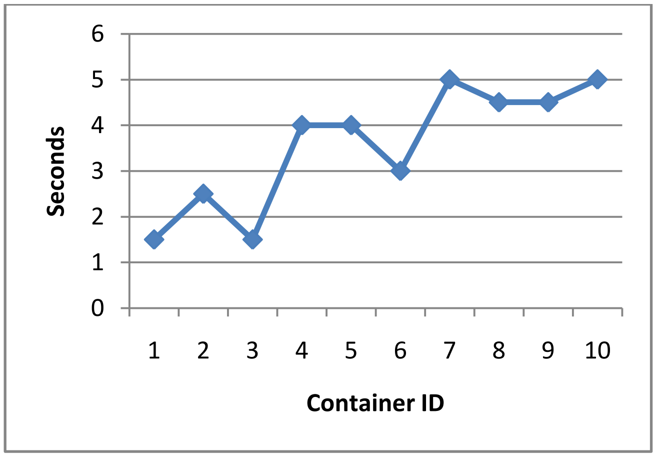


Fig 7. Time for starting a new container.

As shown in fig.7 the horizontal axis is represented as container ID while the vertical axis represented as seconds.

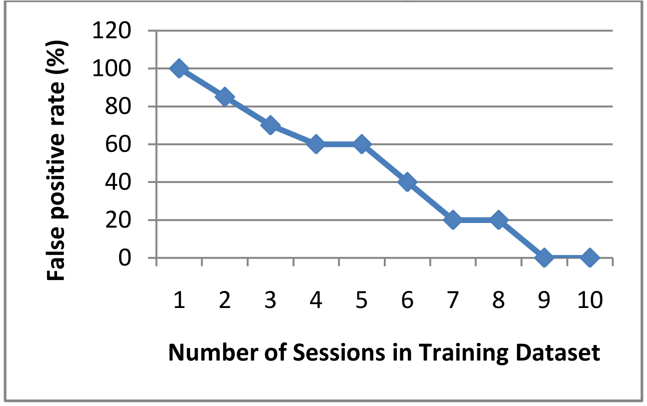


Fig 8. False positives versus training time in static website.

As shown in fig.8 the horizontal axis is represented as number of sessions in training dataset while the vertical axis represented as false positive rate.

VI. CONCLUSION

The existing intrusion detection systems focus either on web server security of database server security. They could not cover the whole application that includes web tier and data tier. Le et al. [10] presented a distributed intrusion detection system that can protect the system from attacks made through front end and also backend. In this paper we built a prototype web application to test the mechanisms to protect the complete web application both at server side and database side. Our intrusion sensors can detect wide range of attacks and prevent them. We built a model to correlate HTTP request to corresponding SQL commands to be executed in database server. We built security mechanisms to prevent attacks such as privilege escalation attack, hijack future session attack, injection attack and direct database attack. The experimental results reveal that the proposed system is capable of protecting the whole web application in distributed fashion.

REFERENCES

[1] "Five Common Web Application Vulnerabilities," http://www. symantec.com/connect/articles/five-common-web applicationvulnerabilities, 2011.

[2] "Common Vulnerabilities and Exposures," http://www.cve. mitre. org/, 2011.

[3] SANS, "The Top Cyber Security Risks," http://www.sans.org/ top-cyber-security-risks/, 2011.

[4] A. Schulman, "Top 10 Database Attacks," http://www.bcs.org/ server.php?show=ConWebDoc.8852, 2011.

[5] Y. Shin, L. Williams, and T. Xie, "SQLUnitgen: Test Case Generation for SQL Injection Detection," technical report, Dept. of Computer Science, North Carolina State Univ., 2006.

[6] C. Anley, "Advanced Sql Injection in Sql Server Applications," technical report, Next Generation Security Software, Ltd., 2002.

[7] J. Newsome, B. Karp, and D.X. Song, "Polygraph: Automatically Generating Signatures for Polymorphic Worms," Proc. IEEE Symp. Security and Privacy, 2005.

[8] H.-A. Kim and B. Karp, "Autograph: Toward Automated Distributed Worm Signature Detection," Proc. USENIX Security Symp., 2004.

[9] 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.

[10] Meixing Le, Angelos Stavrou and Brent ByungHoon Kang, "DoubleGuard: Detecting Intrusions in Multitier Web Applications", VOL. 9, NO. 4, JULY/AUGUST 2012.

[11] Openvz, http://wiki.openvz.org, 2011.

[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] M. Roesch, "Snort, Intrusion Detection System," http://www. snort.org, 2011.

[14] 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.

[15] M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna, "Swaddler: An Approach for the Anomaly-Based Detection of State Violations in Web Applications," Proc. Int'l Symp. Recent Advances in Intrusion Detection (RAID '07), 2007.
[16] C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), Oct. 2003.
[17] 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.
[18] A. Srivastava, S. Sural, and A.K. Majumdar, "Database Intrusion Detection Using Weighted Sequence Mining," J. Computers, vol. 1, no. 4, pp. 8-17, 2006.
[19] S.Y. Lee, W.L. Low, and P.Y. Wong, "Learning Fingerprints for a Database Intrusion Detection System," ESORICS: Proc. European Symp. Research in Computer Security, 2002.
[20] 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.
[21] 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.
[22] greensql, http://www.greensql.net/, 2011.
[23] 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.
[24] D. Wagner and D. Dean, "Intrusion Detection via Static Analysis," Proc. Symp. Security and Privacy (SSP '01), May 2001.
[25] M. Christodorescu and S. Jha, "Static Analysis of Executables to Detect Malicious Patterns," Proc. Conf. USENIX Security Symp., 2003.
[26] V. Felmetsger, L. Cavedon, C. Kruegel, and G. Vigna, "Toward Automated Detection of Logic Vulnerabilities in Web Applications," Proc. USENIX Security Symp., 2010.
[27] Linux-vserver, http://linux-vserver.org/, 2011.
[28] "Virtuozzo Containers," http://www.parallels.com/products/ pvc45/, 2011.
[29] Y. Huang, A. Stavrou, A.K. Ghosh, and S. Jajodia, "Efficiently Tracking Application Interactions Using Lightweight Virtualization," Proc. First ACM Workshop Virtual Machine Security, 2008.
[30] B. Parno, J.M. McCune, D. Wendlandt, D.G. Andersen, and A. Perrig, "CLAMP: Practical Prevention of Large-Scale Data Leaks," Proc. IEEE Symp. Security and Privacy, 2009.

BIOGRAPHIES

Pravallika.P is student of MallaReddy College of Engineering and Technology, Hyderabad, AP, INDIA. She has received B.Tech Degree Computer Science and Engineering and M.Tech Degree in Computer Science and Engineering. Her main research interest includes Networking and Datamining.

R.Radha is working as an Associate Professor in MRCET JNTUH, Hyderabad, and Andhra Pradesh, India. She has completed M.Tech (C.S.E) from JNTUH. Her main research interest includes Networking and WSN.