Double Guard: Detecting Intrusion in Multitier Web Application

Charu Nair[1] Rekha Chauhan[2] Raina Ritu[3]
[1,2,3]Department of Information Technology
[1,2,3]DYPIET, Pimpri, Pune, India

Abstract— Nowadays web services and applications have become an integral part of our life which lets us communicate, store and handle our data from any part of the world. These services and applications are provided through web servers which use database server. These web servers are vulnerable to attacks. Hence, it becomes necessary to prevent an unauthorized access that threats the confidentiality, integrity and authority of an user. Tools available in the market individually detect the abnormal network traffic sent either to the web IDS and database IDS. In proposed system, Double Guard an IDS , which monitors both web and subsequent database requests by modeling the network behavior of user sessions which prevents attacks that an independent IDS would not identify. The proposed system uses Apache web server with MYSQL and Lightweight Virtualization.

Key words: Intrusion Detection, Modeling Technique, Virtualization, Multitier Web Application

I. INTRODUCTION

Over the past decade or so, the web-delivered services and applications has been embraced by millions of businesses as an inexpensive channel to communicate and exchange information. These services use web server front end to run the application user interface and back-end server for database. Many of these databases contain valuable information making them a frequent target of hackers. These hackers prefer gaining access to the sensitive data residing on the database server because of the immense pay-offs in selling the data. In order to prevent such activities many Intrusion Detection Systems (IDSs) have been introduced .These IDSs examine network packets individually within both the web server and database server.

There are different ways in which IDSs can be classified, like active and passive IDS, network-based and host-based IDS and knowledge-based and behavior-based. An active IDS (also known as Intrusion Prevention System-IPS) is a system that automatically block suspected attacks in progress without any intervention required by an operator. It is placed in-line along a network boundary, thus which makes it susceptible to attack. The passive IDS monitors and analyze network traffic activity and alert an operator to attacks and potential vulnerabilities. It isn't capable of performing any protective or corrective function on its own. A knowledge-based or signature-based IDS references a database of previous attack profiles and known system vulnerabilities to identify active intrusion attempts. Also behavior-based or statistical anomaly-based IDS references a baseline or learned pattern of normal system activity to identify active intrusion attempts and to protect multi-tiered web services. The IDS on the web and the database can detect abnormal network traffic sent to either of them. However, it is found that there are cases where the normal traffic is used to attack the web-server and the database-server which cannot be detected by these IDSs.

There also exist many IDS including Snort, GreenSQL, Suricata, OpenDLP, Kismet and many more. GreenSQL is an Open Source database firewall used to protect databases from SQL injection attacks. GreenSQL works in a proxy mode and has built in support for MySQL**.** The logic is based on evaluation of SQL commands using a risk scoring matrix as well as blocking known db administrative commands (DROP, CREATE, etc). GreenSQL[7] is distributed under the GPL license. But it is unable to detect attacks like privilege escalation attack, web server aimed attack and direct database attack. Snort, an open source network IDS is a packet sniffer that monitors network traffic in real time, scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies. It is unable to detect SQL injection attack, privilege escalation attack and DirectDB attack. Suricata is also an open source based intrusion detection system. It was developed by the Open Information Security Foundation. It is a high performance Network IDS. Suricata is interchangeable with snort and can be used in the place of snort with minimal work. It is also multithreaded. OpenDLP is a free and open source, agent and agentless based, centrally managed, massively distributable data loss prevention tool. But the limitations is that in order to deploy to multiple systems with a single profile ,you must have domain admin credentials but if you don't have domain admin credentials, you need to create profile for each system with different passwords. Kismet is a network detector, packet sniffer and IDS for 802.11 wireless LANs. It works with any wireless card which supports raw monitoring mode. But it has the complexity issues, reiterations and level specific.

In this paper, we present DoubleGuard, which is used to detect attacks in multitiered web services. This approach can create normality models of isolated user sessions that include both the web front-end (HTTP) and back-end (File or SQL) network transactions. To achieve this, a light-weight virtualization technique is employed to assign each users web session to a dedicated container, an isolated virtual computing environment. The container ID is used to accurately associate the web request with the subsequent DB queries. Thus, DoubleGuard can build a causal mapping profile by considering both the webserver and DB traffic.

Different methods used in this system are such as Virtualization technique, Invariants technique, Clustering technique, case-based reasoning technique, library interposition technique. Hash function algorithm such as MD5 is used.

This paper is divided into many sections like: section [II] represents related work ,section[III] represents proposed methodology, section[IV] represents results and discussion and section[V] represents conclusion and futurescope.

II. RELATED WORK

To bring the ideas of double guard we have mentioned different papers.

The author Odeh, S.M[1] discusses signature verification and recognition using a new approach which depends on a neural network which enables the user to recognize whether a signature is original or fraud. The user introduces into the computer the scanned images, modifies the quality of image by image enhancement and noise reduction techniques, that is to be followed by feature extraction and neural network training, and finally verifies the authencity of the signature. But in this most of the dynamic information in the signature is lost.

The author Srihari, S.N[2] identifies that Learning strategies and classification methods for verification of signatures from scanned documents are proposed and evaluated. They are considered as writer independent- those that learn from a set of signature sample prior to entry of a writer, and writer dependent- those that learn only from a newly enrolled individual. Classification methods include two distance based methods (one based on a threshold, the standard method of signature verification, and the other based on a distance probability distribution), a Nave Bayes classifier based on pairs of feature bit values and a support vector machine (SVM). Two scenarios are considered for the writer dependent scenario: (i) without forgeries (one-class problem) and (ii) with forgery samples being available (two class problem). In the one-class scenario distance methods are superior while in the two-class problem SVM based method outperforms the other methods.

The author Rathod,Y.A[4] describes an approach for intrusion detection system for database management system.The approach concentrate on security policies for transactions permitted with DBMS.The approach is designed to mine audit log of legitimate transaction performed with database and generate signature for legal transactions as per security policy.The transactions not complaint to signature of valid transactions are identifies as malicious transactions.

The author sriraghavan,R.G.[3] introduces an anomaly detection system for web-based applications. The anomaly detection system monitors the attribute value pairs of successful HTTP requests received by webserver applications and automatically creates parameter profiles. It then uses these profiles to detect anomalies in the HTTP requests. Customized profiles help reduce the number of false positives. Automatic learning ensures that the system can be used with different kinds of web application environments, without the necessity for manual configuration. But in this, malicious activity that falls within normal usage patterns is not detected & it is difficult to define rules.

The author Wentao Liu[5] discusses principles of DOS attack and some DoS attack methods are deeply analyzed. The DoS attack detection technologies which include network traffic detection and packet content detection are presented. The DDoS based on DoS is introduced and some DDoS tools are described and the important TCP flood DoS attack theory is discussed. The DoS attack program and a DoS attack detection program based on Winpcap for experiment are designed and the network packet generation and capture are implemented. The experiment expressed the key progress of DoS attack and detection in detail.

The author Dewangan and Agrawal[6] studies the avalanche effect in advanced encryption standard using binary codes. In cryptography, the avalanche effect refers to a desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions. Avalanche Effect is calculated by changing one bit in plaintext keeping the key constant and by changing one bit in encryption key keeping the key constant. The avalanche effect is evident if, when an input is changed slightly ,the output changes significantly.

Since databases always contain more valuable information,
significant research efforts have been made on database IDS and database firewalls. The software such as Green SQL is a database firewall engine used to protect open source databases from SQL injection attacks. It works in proxy mode.

Instead of connecting to a database server, web applications will first connect to a database firewall. SQL queries are analyzed; if they are found safe, then they are forwarded to the back-end database server.

CLAMP[8] is an architecture for preventing data leaks even in the presence web server or SQL injection attacks. It protects sensitive data by enforcing strong access control on user data and by isolating code running on behalf of different users. By focusing on minimizing developer effort, we arrive at an architecture that allows developers to use familiar operating systems, servers and scripting languages, while making relatively changes to application code.

III. PROPOSED METHODOLOGY

For efficient understanding of proposed system we need to understand the procedure for working of third party data handling scenario. A third party service provider is an organization which actually stores and protects many clients data upon same mutual agreement. So, there will be a huge threat for the data from the internal employees of third party. So, third party service provider need to enhance the system of his security to protect the data from tampering and algorithm need to identify the culprits who succeed in tampering . So this completes scenario is perfectly coupled with our system and which can be shown in fig. 1

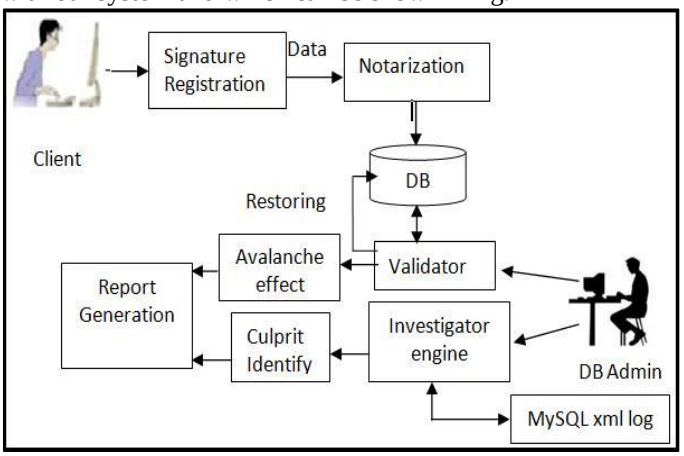

Fig. 1: Overview of Tamper Detection System in Database

Our proposed method can be easily describe with following steps:

1) Step 1: Here all the clients who are interested to outsource their database for restoring purpose need register at third party. This registration is empowered signatures creation and registration along with the profile creation . This signature creation will be done using strong one way hashing algorithm like MD5.On saving every record of data on third party side by the client system ensures the right source of the clients using the unique signature which is provided to client by the system.
2) Step 2: Here a service called notarization is actually participate in finding the right source of data on its arrival from the client . This is done by comparing signatures of the client with its original signature which store on profile creation. If the signatures are match then only data from the client are allowed to store third party database server. So, now the data which is store at third party can be considered as original and generated from right source.
3) Step 3: A special subsystem is deployed to identify the tampered id in the database called a validator. Here validator is assigned a time for visiting the database in regular intervals. For each visit of the database all the records are fetched and processed in the form of hash keys formed by MD5 algorithm. These hash key sets are compared with the previous visit set for the intrusion for the assigned time of validator. Then the changes in a single bit of hash key reflects a greater change in the database. This is represents as an "Avalanche effect". Once the "Avalanche effect" is been identify in the database then every record in the database are linearly compared with the previous one to identify the exact attributes of tampered data. This process is enforced with recursive multithreading that efficiently handles the tampered detection process even in smaller time of validation. This can be shown in algorithm 1
4) After detection of which and what now our system is keen to identify who and when of the tampering. This is done by a heuristic approach where another a validation is triggered simultaneously for the assigned time. This validator actually keeps an eye on MySQL xml log to identify the culprit name on detection of tampering in database by the step 3.

   Once the culprit name is extracted from xml log immediately system extracted the tamper date and time and report all in a well structured manner.
5) Once the system successfully detects what, who and when of tampering then the main issue is remind the system is about the loss of the data. To recover this loss the data which is carried by the validator in step 3 in its previous visit is actually the original data. This is been restore again in the database for the tampered id to make up the loss. This feature of our system always makes the client to keep the data at third party without any doubts.

Algorithm 1

//Input:- T as Table name T as
Time interval
//Output:- Tampered id

0: Start
1: Get database as DB and table name as T
2: Visit table T_v at interval T
3: T_v contain in double dimension m_{ij} (master vector) 4:
for i= 0 to m_{ij} size at T
5: **for** j=0 to m_{ij} size at T-1
6: Get DB tuple as D_{iT} and D_{T-1}
7: Apply MD5 hash on D_{iT} and D_{T-1}
(D_T != $D_T\delta_1$ **if**)
9: **then** detect tamper
10: Stop inner for
11: Stop outer for
12: Stop

Fig. 2: Algorithm 1

IV. Results and Discussions

To show the effectiveness of proposed system some experiments are conducted on java based windows machine. To measure the performance of the system we set the bench mark by selecting three different attacks like Sql injection, DoS attack and Data modification attack in real world scenario in our model.

To determine the performance of the system, we examined how many relevant Attacks are identified based on our double guard approach.

To measure this precision and recall are considering as the best measuring techniques. So precision can be defined as the ratio of the number of relevant attacks identified to the total number of irrelevant and relevant attacks identified. It is usually expressed as a percentage. This gives the information about the relative effectiveness of the system.

Whereas Recall is the ratio of the number of relevant attacks are identified to the total number of relevant attacks identified. It is usually expressed as a percentage. This gives the information about the absolute accuracy of the system.

The advantage of having the two for measures like precision and recall is that one is more important than the other in many circumstances.

For more clarity let we assign

- A = The number of relevant attacks identified,
- B = The number of relevant attacks not identified, and
- C = The number of irrelevant attacks identified.

So, Precision = (A/ (A+ C))*100
And Recall = (A/ (A+ B))*100

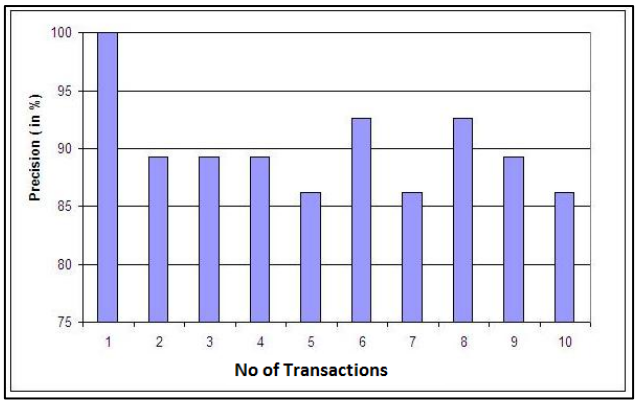

Fig. 3: Average Precision of the Proposed Approach

In Fig. 3, we observe that the tendency of average precision for the identified attacks are high compared to other systems.

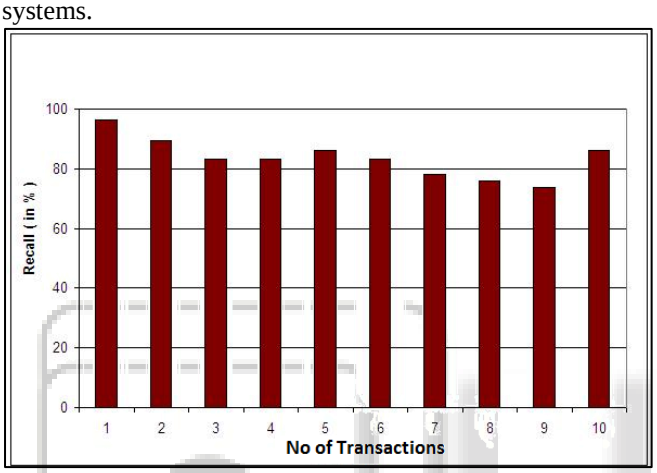

Fig. 4: Average Recall of the Proposed Approach

In Fig. 4, we observe that the tendency of average Recall for the identified attacks are high compared to other system. So this shows that our proposed system is achieving high accuracy than any other method.

V. Conclusion and Future Scope

Web services contains different layers and data is transferred from the user to the database layer. During this transformation many errors are occurred which can be detected by the IDS system i.e Double Guard. Double guard is a system that builds models of normal behavior for multi-tired web applications from both front-end web requests and back-end database queries. It forms container based IDS with multiple input streams to produce alerts. Double guard was able to identify a wide range of attacks with minimal false positives. In future work we are proposing to derive an efficient system to detect all the attacks.

References

[1] Odeh, S.M and Khalil M, " offline signature verification and recognition:neural network approach" , Innovations in Intelligent Systems and Applications (INISTA),2011.

[2] Srihari, S.N, kalera,M.K," learning strategies and classification methods for off-line signature verification", Frontiers in Handwriting Recognition,2004.

[3] Sriraghavan, R.G and lucchese ," data processing and anomaly detection in web based applications", Machine Learning for Signal Processing ,2008.

[4] Rathod,Y.A , Chaudhari," database intrusion detection by transaction signature", Computing Communication & Networking Technologies, 2012.

[5] Wentao Liu and Wuhan, "Research on DoS Attack and Detection Programming",Intelligent Information Technology Application,2009.

[6] Dewangan, Agrawal, Mandal and tiwari, "Study of avalanche effect in AES using binary codes", Advanced Communication Control and Computing Technologies(ICACCCT),2012.

[7] greensql. http://www.greensql.net/.

[8] B. Parno, J. M. McCune, D. Wendlandt, D. G. Andersen, and A. Perrig."CLAMP: Practical prevention of large-scale data leaks.", In IEEE Symposium on Security and Privacy. IEEE Computer Society, 2009.

