Double Guard: Detecting Intrusion in Multitier Web Application

Abhaya Raut, Aditi Shah, Chaitrali Pande, Prof. Arya C.S.

Abstract— **For enabling communication and the personal information management from anywhere, an inextricable part of daily life is the Internet services and applications. To entertain this increase in application and complexity of data, web services have moved to a multi-tiered design wherein the application front-end logic runs by the web server and to a database or file server the data are outsourced. In this system, we propose an IDS system which is Double Guard, that models the network user sessions behaviour across both the back-end database as well as the front-end web server. By monitoring both web and subsequent database requests we are able to ferret out attacks that independent IDS would not be able to identify. Furthermore, in terms of sessions of training and functionality coverage we measure any multitier IDS limitations. With My SQL and lightweight virtualization with the help of an Apache web server we implement Double Guard. In both dynamic and static web applications the real-world traffic then processed and collected over a 15-day of system deployment period. Finally, using Double Guard, we will able to display a wide range of attacks which will give the 100 percent accuracy while for web services which are static, 0 percent false positives maintain by it and for web services which is dynamic, it is 0.6 percent false positives.**

Index Terms— IDS system, Web Services, HTTP Request, Anomaly Detection

I. INTRODUCTION

To protect multi-tiered web services, Intrusion detection systems (IDS) have been widely used to detect known attacks by matching misused traffic patterns or signatures [1]. In the existing system we require different IDS one for web server and another for database server. Two IDSes required so we need to create two IDSes with different prevention measure first IDS that contains prevention measure related to web server so attack should not happen on web server but some time attack happen on database server by passing web server so for that reason need to create another IDS with prevention measure related to database server attack. We want to avoid creating two IDS so we are creating one DoubleGuard system that act as IDS and prevent both side of attack. Attack may be on web server or database server. Most of the IDS examine the attack individually on web server and database server. In order to protect multi-tiered web services an efficient system call Intrusion Detection System is needed to detect attacks by mapping web request and SQL query, there is direct causal relationship between request received from the front end web server and those generated for the database backend. We It will allocate the isolated session for each user it is practical for most of the web applications. Static web sites has the controlled environment whereas dynamic website not. Dynamic web site allow persistent back end data modification through the HTTP requests to include the parameters that are variable and depend on the user input. Because of which the mapping between the web and the database rang from one to many as shown in the mapping model. In the proposed system we are implementing Double Guard that handle both sides of attack. Attack may be from static web site or dynamic web site. No need to create two different IDSes for two different web site. Double Guard can handle both types of attack.

Following tasks should be accomplished by DoubleGuard:

- It should prevent the damage that detected intrusion could cause
- It should mitigate the damage that detected intrusion could cause to discover the new attacks patterns
- Accuracy: It must not identify the legitimate action in system environment as anomaly or misuse like IDS
- Performance: DoubleGuard performance must be high enough to carry out the real time intrusion detection
- Completeness: It should not fail to detect an intrusion. It is practically impossible because it is impossible to have a global knowledge about past, present and future attacks.
- Fault tolerance : It should be resistant to attacks and its consequences
- Timeliness: DoubleGuard should perform the analysis as quickly as possible.

Apart from the functional requirements, DoubleGuard should satisfy the number of economical requirements, in particular case, cost.

- Cost of the product
- Cost of additional computer resources needed
- Cost of administration
- An importance of all this is oblivious. DoubleGuard should available not only to large enterprises, but also small enterprises, as well as private person.

II. LITERATURE SURVEY

While developing any system it is necessary to do literature survey of which are existing systems available in market. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples.

While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazons S3 is such an example. CSPs have a feature of storing users data in the form of chunks for recovery. Due to distributed architecture, chunks are stored at different position from the original file. In existing system, client has an direct access to that location whenever he makes an request for file recovery. Thus, there is a possibility of files being corrupt due to clients mistakes. So in proposed system, we have eliminated this problem by denying clients direct access to the chunks location.

Meixing Le,Angelos Stavrou, Brent ByungHoon Kang,"DoubleGuard: Detecting Intrusions In Multi-tier Web Applications" proposed A network Intrusion Detection System (IDS) can be classified into two types: anomaly detection and misuse detection. Anomaly detection first requires the IDS to define and characterize the correct and acceptable static form and dynamic behavior of the system, which can then be used to detect abnormal changes or anomalous behaviors. The boundary between acceptable and anomalous forms of stored code and data is precisely definable. Behavior models are built by performing a statistical analysis on historical data, or by using rule-based approaches to specify behavior patterns. An anomaly detector then compares actual usage patterns against established models to identify abnormal events. Our detection approach belongs to anomaly detection, and we depend on a training phase to build the correct model. As some

Legitimate updates may cause model drift, there are a number of approaches that are trying to solve this problem. Our detection may run into the same problem; in such a case, our model should be retrained for each shift. Intrusion alerts correlation provides a collection of components that transform intrusion detection sensor alerts into succinct intrusion reports in order to reduce the number of replicated alerts, false positives, and non-relevant positives. It also fuses the alerts from different levels describing a single attack, with the goal of producing a succinct overview of security-related activity on the network. It focuses primarily on abstracting the low-level sensor alerts and providing compound, logical, high-level alert events to the users. DoubleGuard differs from this type of approach that correlates alerts from independent IDSes. Rather, DoubleGuard operates on multiple feeds of network traffic using a single IDS that looks across sessions to produce an alert without correlating or summarizing the alerts produced by other independent IDSs. An IDS also uses temporal information to detect intrusions. DoubleGuard, however, does not correlate events on a time basis, which runs the risk of mistakenly considering independent but concurrent events as correlated events. DoubleGuard does not have such a limitation as it uses the container ID for each session to causally map the related events, whether they be concurrent or not.

Niraj Gaikwad, Swapnil Kandage, Dhanashri Gholap,"DoubleGuard: Detecting Preventing Intrusions in Multitier Web Applications" proposed Intrusion detection systems have been widely used to detect the attacks which are known by matching misused traffic patterns or signatures to protect the multi tiered web services. The IDS class has a power of machine learning which can detect unknown attack by identifying the abnormal behavior of the network traffic action from previous behavior of IDS phase. The abnormal network traffic which are send by the attacker to attack the server can be detected by the web IDS and the database IDS and prohibit to enter within the server. But, if the attacker uses the normal traffic to attack the web servers and database server then such type of attack cannot be able to detect by a IDSs.

K.Karthika, K.Sripriyadevi,"To Detect Intrusions in Multitier Web Applications by using Double Guard Approach." proposed In DoubleGuard, the new container-based webserver architecture enables us to separate the different information flows by each session. For the static webpage, our DoubleGuard approach does not require application logic for building a model. However, as we will discuss, although we do not require the full application logic for dynamic web services, we do need to know the basic user operations in order to model normal behavior. DoubleGuard focuses on modeling the mapping patterns between HTTP requests and DB queries to detect malicious user sessions. Building the mapping model in DoubleGuard would require a large number of isolated web stack instances so that mapping patterns would appear across different session instances.

III. EXISTING SYSTEM

In existing system individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them.

These IDSs cannot detect cases wherein normal traffic is used to attack the webserver and the database server.

For Example if an attacker with non admin privileges can log in to a webserver using normal-user access credentials, he/she can find a way to issue a privileged database query by exploiting vulnerabilities in the web server.

IV. PROBLEM STATEMENT

Lots of existing intrusion Detection System (IDSs) examines the network packets individually within both the web server and the database system. However, there is very little work being performed on multi-tiered Anomaly Detection (AD) systems that generates models of network behavior for both web and database network interactions. In such multi-tiered architectures, the back-end database server is often behind a firewall while the web servers are remotely accessible over the internet. Unfortunately, though they are protected from direct remote attacks, the back end systems are susceptible to attacks that use web requests as a mean to exploit the back end. In order to protect multi-tiered web services, an e-client system call as Intrusion detection system is needed to detect known attacks by matching misused traffic patterns or signature.

V. MATHEMATICAL MODEL

A. Admin login

Set (F)={f0,f1,f2,f3}
F0-Intrested user
F1-Personal information
F2- Id and Password
F3- enables protection

B. New user registration

Set (M)={m0,m1,m2,}
M0- Interested new user
M1- New user details
M2- Id and Password
M3-New user login

C. Decision Making Module

Set (D)={m3,d0,d1,d2,d3}
D0- adopt anomaly-based detection mechanism
D1-analyze the different attack
D2-update the attack signature database
D3- distinguish particular attacks from legitimate traffic

D. Evaluation of Attack detection

Set (E)={e0,e1,e2}
E0-filter the records
E1-finalize the type of attack
E2-Evaluate the result

VI. PROPOSED WORK

We present Double-Guard, an IDS system that models the network behavior of user sessions across both the front-end web server and the back-end database. By monitoring both web and subsequent database requests, we are able to ferret out attacks that independent IDS would not be able to identify. Furthermore, we quantify the limitations of any multitier IDS in terms of training sessions and functionality coverage. Using Double-Guard, we were able to expose a wide range of attacks.

In proposed System we are going to use light weight container which can map the request for web server and query to the DB server. Double Guard can build a causal mapping profile by taking both the webserver and DB traffic into account. We have design an algorithm which generate unique key for each HTTPS request and DB query.

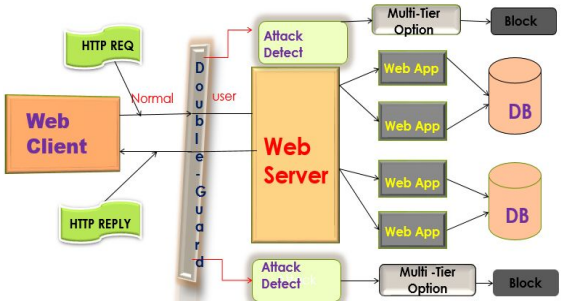

Fig 1. System Architecture

Advantages of Proposed System

1. The proposed IDS will help to detect the intrusion on both front end (Web Server) and back end (Database Server).
2. Double Guard able to identify a wide range of attacks with minimal false positives.

In this project, to detect attacks in web services which are multi-tiered, the present Double Guard system is used. In this system, isolated user sessions normality models can create that include both the web front end (HTTP) as well as back end (File or SQL) transactions of network. For achieving this a technique which is light weight virtualization technique is used for assigning the web session of every users to a particular container, an environment of isolated virtual computing.

VII. ALGORITHM IMPLEMENTATION

In this algorithm we are getting set of web request and gene rate SQL query according to web request. If user perform web request and for that web request SQL query is not generated then that web request mark as EQS (Empty Query Set) else generated SQL query and get result. If we got same result as expected up to threshold value then mapping is correct otherwise need more training sessions. In NMR (No match request) SQL query genera ted without web request from user but according to SQL query action will be performed. We have used the following static model building algorithm to create the static webpage.

Algorithm 1: Static Model Building Algorithm.

Require: Training Dataset, Threshold t
Ensure: The Mapping Model for static website
1: for each session separated traffic Ti do
2: Get different HTTP requests r and DB queries q in this session
3: for each different r do
4: if r is a request to static file then
5: Add r into set EQS
6: else
7: if r is not in set REQ then
8: Add r into REQ
9: Append session ID i to the set ARr with r as the key
10: for each different q do
11: if q is not in set SQL then
12: Add q into SQL
13: Append session ID i to the set AQq with q as the key
14: for each distinct HTTP request r in REQ do
15: for each distinct DB query q in SQL do
16: Compare the set ARr with the set AQq
17: if ARr = AQq and Cardinality(ARr) > t then
18: Found a Deterministic mapping from r to q
19: Add q into mapping model set MSr of r
20: Mark q in set SQL
21: else
22: Need more training sessions
23: return False
24: for each DB query q in SQL do
25: if q is not marked then
26: Add q into set NMR
27: for each HTTP request r in REQ do
28: if r has no deterministic mapping model then

29: Add r into set EQS
30: return true .

VIII. RESULT

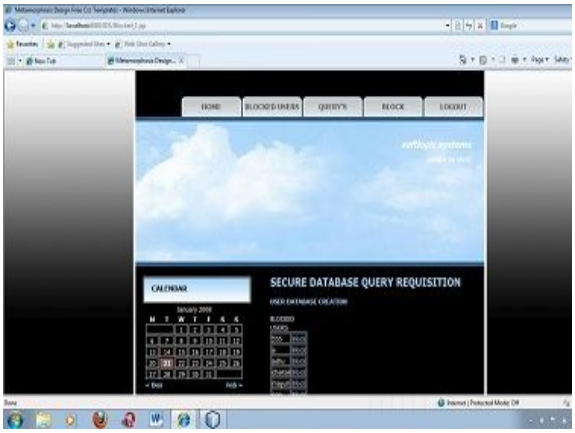

Fig 1. Main Page

Fig 2. Query Registration

Fig 3. Login Page

IX. CONCLUSION

In this system, we are developing the Double guard an Intrusion Detection System which manages both front and back end of the multi-tier design & exposes a wide range of attacks with 100% accuracy. DoubleGuard is a system used to detect attacks in multi-tiered 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. With a lightweight virtualization, from each web server session this project accomplish this by confine the flow of information. Furthermore, we quantified the accuracy of detection in our approach when we try to model web requests of static as well as dynamic with the back end file system and queries of database.

ACKNOWLEDGMENT

Authors thanks to project guide Prof. Arya C.S. who always being with presence & constant, constructive criticism to made this paper. We would also like to thank all the staff of COMPUTER DEPARTMENT for their valuable guidance, suggestion and support through the project work, who has given co-operation for the project with personal attention. Above all we express our deepest gratitude to all of them for their kind-hearted support which helped us a lot during project work. At the last we thankful to our friends, colleagues for the inspirational help provided to us through a project work.

REFERENCES

[1] Meixing Le, Angelos Stavrou, Brent ByungHoon Kang." DoubleGuard: Detecting Intrusions In Multi- tier Web Applications" IEEE transaction on dependable and secure computing vol.9 no.4 year 2012

[2] Niraj Gaikwad, Swapnil Kandage, Dhanashri Gholap, "DoubleGuard: Detecting & Preventing Intrusions in Multitier Web Applications", International Journal of Networks and Systems, of Networks and Systems, 2(2), February – March 2013, 09 – 14, ISSN 2319 – 5975.

[3] Rahul Dandwate, Lomesh Ahire, Dipali Kumbhar, Pratik Kamble, Aniket Shirude, Shweta Bhandakkar, "DOUBLEGUARD: DETECTING INTRUSIONS IN MULTITIER WEB ARCHITECTURE", Proceedings of IRF International Conference, 13th April-2014, Pune, India, ISBN: 978-93-84209-04-9

[4] K.Karthika, K.Sripriyadevi ," To Detect Intrusions in Multitier Web Applications by using Double Guard Approach" , International Journal of Scientific & Engineering Research Volume 4, Issue 1, January-2013 1 ISSN 2229-5518.

[5] 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.

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

[7] 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.

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

[9] R. Sekar, "An Efficient Black-Box Technique for Defeating Web Application Attacks," Proc. Network and Distributed System Security Symp. (NDSS).

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

[11] G. Vigna, W.K. Robertson, V. Kher, and R.A. Kemmerer, "A StatefulIntrusion Detection System for World-Wide Web Servers," Proc. Ann. Computer Security Applications Conf. (ACSAC '03), Oct.2003.