ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

# Intrusion Detection Using Double Guard In Multi-Tier Architecture

K.Kavitha[1], S.V.Anandhi[2]

Student, M. E., Dept. of CSE, Dr.SACOE, Thiruchendur, Tamilnadu, India[1]

Associate Professor, Dept. of CSE, Dr.SACOE, Thiruchendur, Tamilnadu, India[2]

**ABSTRACT**— Due to internet expansion web applications have now become a part of everyday life. To put up this increase in application and data complexity, web services have moved to a multi-tiered design wherein the web server runs the application front-end logic and data is outsourced to a database or file server. In this paper, we designed IDS in multitier architecture that models the network behavior of user sessions across both the front-end web server and the back-end database. It handles the relations between the actions caused by the user for every simple user sessions and is also designed to detect potential violations in database security. This system is implemented using Apache web server with MySQL and lightweight virtualization. For each client a "Web Server Virtual Machine" is created and is associated with an independent container ID and hence it enhances the security. The concept of holder and the user behavior pattern provides a means of tracking the information flow from the web server to the database server for each session.

**KEYWORDS**-- Multitier Web Application, Virtualization, Web-Based Attacks, container-based architecture.

## I. INTRODUCTION

Internet services have been speedily expanded across the globe in relation to its popularity and complexity. The World Wide Web provides varied applications such as the social media, educational, finance etc. But, the wide use of these services has made them reside on the peak of attackers. Different types of Attacks hijacking session of users and direct database attack are carried out on the web servers where the web server is taken over by the attacker. Hence, all the subsequent user sessions are also being hijacked.

The anomaly detection systems detect the abnormal behaviors of the system. The network packets are individually analyzed by the intrusion detection systems. But for the system designed to be multitier a very modest task is carried out. Hence, due to the lack of such architecture even if we make use of firewall to guard the database backend, they are likely to be attacked by the attackers who employ the web request for exploiting the back end data. The misused detection can be used for matching the changed patterns in order to prevent the multi tier web services. But, when the network traffic is transfer from the web to the database or vice-versa, the intrusion detection system is unable to identify the anomalous traffic by just using the web intrusion detection system or the database intrusion detection system. The multitier web application analyzer is being developed for preventing the system from wide number of attacks. Normality Model of the isolated user sessions is being built which consists of the web application front end that is in form of HTTP requests and the data base backend i.e. the SQL or file server. Here, we present a lightweight virtualization model for assigning every user's session to a container. To specifically relate the web request with the consequent database queries we formulate use of the container ID. The IDS builds a casual mapping profile by undertaking the web server and database traffic. The lightweight virtualization helps in running innumerous copies of the web server instances in different containers. Hence, everyone is differentiated from the rest. Each client is assigned a dedicated container so that even when the attacker attacks the session, it is limited to that session only and doesn't damage the other user session.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

## II. EXISTING WORK

Person At present, the Intrusion Detection System is a device or software application that helps to monitoring network and system vulnerabilities. It generates reports for a management station. The anomaly based detection requires the system to describe and organize the right and acceptable form and nature that can be used for finding the behaviors that are different from others.

Another approach is the Intrusion Detection Alert correlation method which involves combination of all security alert events like the replicated alerts, false alarms and other non relevant data to the users. But, in place of correlating the alerts from the individual detection system our methodology works on multiple feed of traffic in the network involving the use of just single IDS.

Another approach Database Intrusion Detection using a data mining technique. Novel weighted data dependency rule mining algorithm using extension of the E-R diagram notations. Role Based Access Control (RBAC) use to finding out sensitive attributes dynamically improve in future.

## III. SYSTEM ARCHITECTURE

3.1 Normality Model Based Architecture

The network traffic from both legal and illegal users is received at an alike web server. The attacker can affect the future sessions by compromising the web server. It is not a legitimate option to allot each session to a dedicated web server, since it may diminish the web server. Hence to gain similar internment when maintaining a low performance and overhead of resources, light-weight virtualization is used. The light-weight process containers that are disposable servers for the client are being used. Thousands of containers can be initialized only on a single machine. These containers can be terminated, reverted or reinitialized for every new client session. Here every session assigned to a dedicated web server is isolated from every other session. The initialization of each virtualized container can be completed using read-only template that is presumed to be clean and hence gives assurity that at initialization every session is being served with a clean web server instance. The communication statement of a single user is expected to go to the alike dedicated web server. Different users can be shown by the sessions to some extent where by permitting us to recognize the suspectable behaviour by the session and user. We will treat the traffic with the session as infected if we sense any abnormal behaviour in a session. The course of information flow is being separated in every container session. This offers us with high security performance. It also maps all the web server requests with the database queries. The normality model shown in fig. 1 helps us to map the relationship between the unauthorised user accesses. The normality model divides the session of the users and thus helps to detect the attacks. It is possible to recognize the web requests and the resulting SQL queries and hence possible to evaluate them in the session. Sensors are placed on both sides of the server which detect anomalies by capturing the traffic at both the end.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

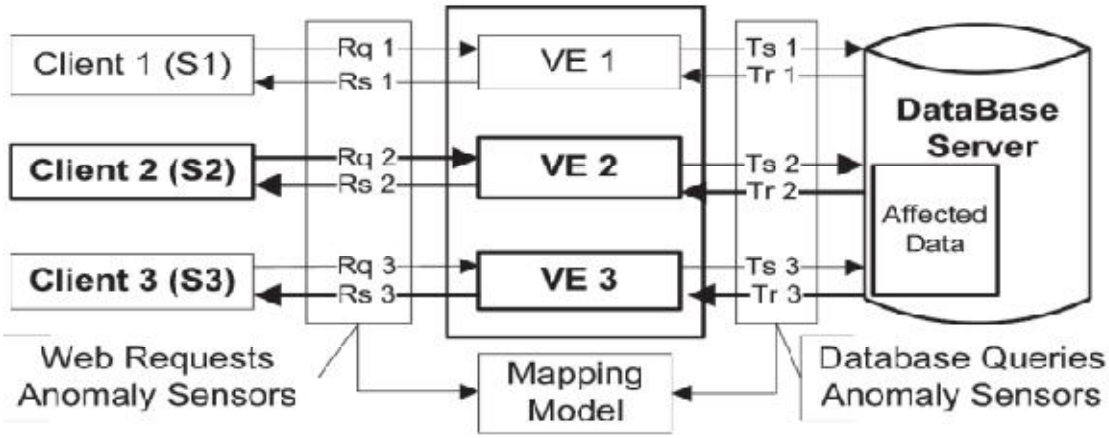


**Figure.1 Normality Model**

**Types of Attacks**

**3.1.1 Hijack Future Session Attack:**

This type of attack happens mostly at the web server side. Attacker invades over the webserver and hijacks all resulting unauthorized user sessions to initiate the attacks. By hijacking other user session for a period of time, the attacker can send or drop replica of messages or replies or even drop user requests. A session hijacking attack can be further categorized as a Spoofing/Man-in-the-Middle attack, an Exfiltration Attack, a Denial-of-Service/Packet Drop attack, or a Replay attack.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

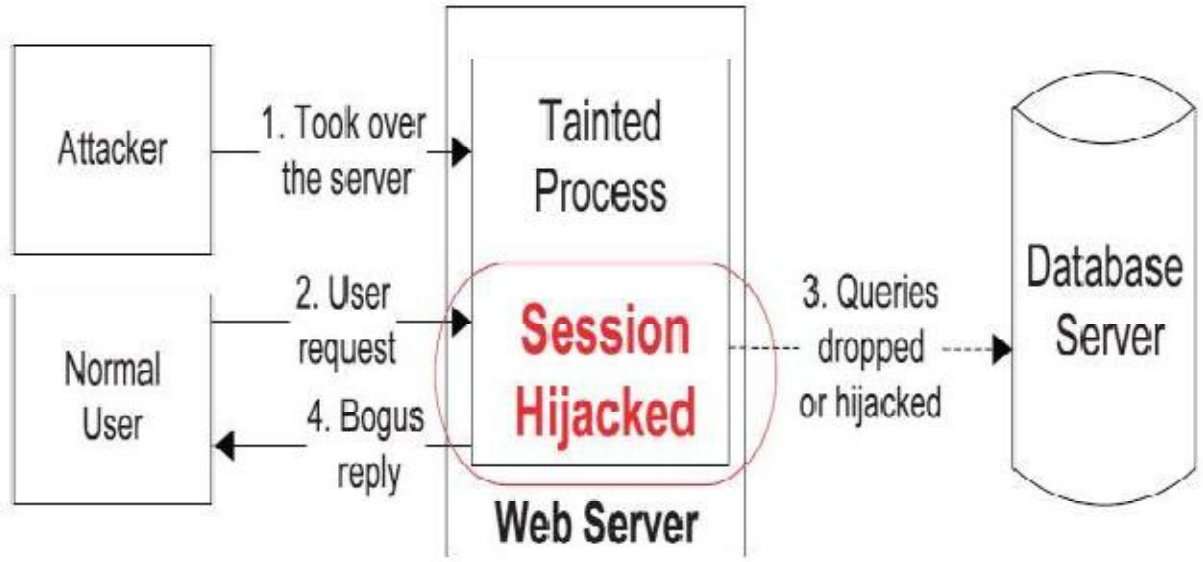


**Figure.2 Hijack Future Session Attack**

According to the mapping model, the web request should invoke some database queries (e.g., a Deterministic Mapping), then the abnormal situation can be detected. However, neither a conventional web server IDS nor a database IDS can detect such an attack by itself. Fortunately, the isolation property of our container-based web server architecture can also prevent this type of attack. As each user's web requests are isolated into a separate container, an attacker can never break into other users' sessions.

### 3.1.2 Direct DB Attack

It is possible for an attacker to bypass the web server or firewalls and connect directly to 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 matched web requests for such queries, a web server IDS could detect neither. Furthermore, if these DB queries were within the set of allowed queries, then the database IDS itself would not detect it either. However, this type of attack can be caught with our approach since we cannot match any web requests queries. In addition, if the database queries are with in the set of valid queries, the database IDS would itself not be able to identify the attack.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

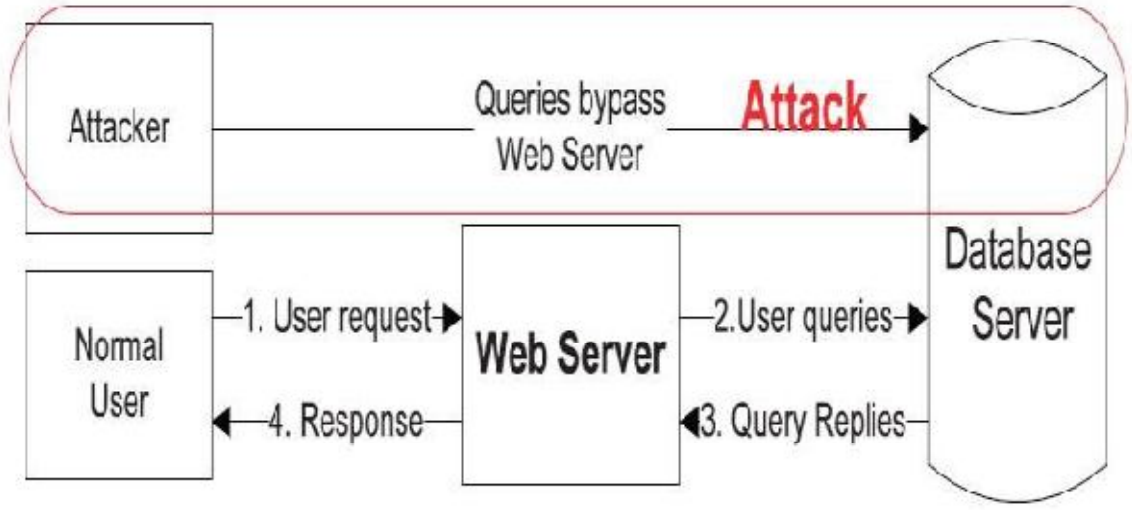


**Figure.3 Direct DB Attack**

## IV. MODELLING DETERMINISTIC MAPPING AND PATTERNS

For static website, the links are static wherein clicking on the same link will always return the same information. Hence we can build an accurate model for mapping relationships between web request and database queries. In place of one-to-one mapping relationship between database queries and web request there can be numerous SQL queries for single web request. For some cases, it may happen such that no queries will be generated by the web request i.e. some of the requests will just retrieve data from web server. On the other hand, one request may let that the web server may invoke number of queries. Hence, we organize the mapping patterns into four classes given below. The request is at the source of the data flow. Consider a set of request rm and set of queries Qp. The total number of sessions in the training phase is N. We can then achieve the set total web request REQ and set of SQL queries SQL across the whole session. The mapping of the model is in form of one request to a query set rm - >Q

### 4.1 Deterministic Mapping

This form is most common type mapping and perfectly matched pattern. Here, the web request rm appears in all traffic with SQL query set Qp. In case, there is an absence of query set Qp for any session in the testing phase with request rm, it indicates that intrusion is present.

### 4.2 Empty Query Set

It is a special case i.e. EQS wherein the SQL query can be an empty set. It indicates that the web request can neither cause nor generate any database queries. The mapping of this kind can be rm->ø. Here web request are put together in the set EQS during the testing phase. Consider an example,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

that when a web request for retrieving a GIF image file is made from the identical webserver, then a mapping relationship doesn't exist as only the web requests are observed.

**4.3 No Matched Request**

Sometimes the queries cannot match-up with the web requests. These unmatched queries are placed in a set NMR. Any query in a set NMR is considered to be legitimate. The NMR size depends on the webserver logic but it is typically small.

**4.4 Non Deterministic Mapping**

The identical web requests can arrive each time. It matches with one query inside the pole of query set. The mapping pattern can be rm -> Qi (Qi € {Qn, Qp, Qq...}). So it is difficult to classify the matched traffic pattern. This happens mostly in case of dynamic websites.

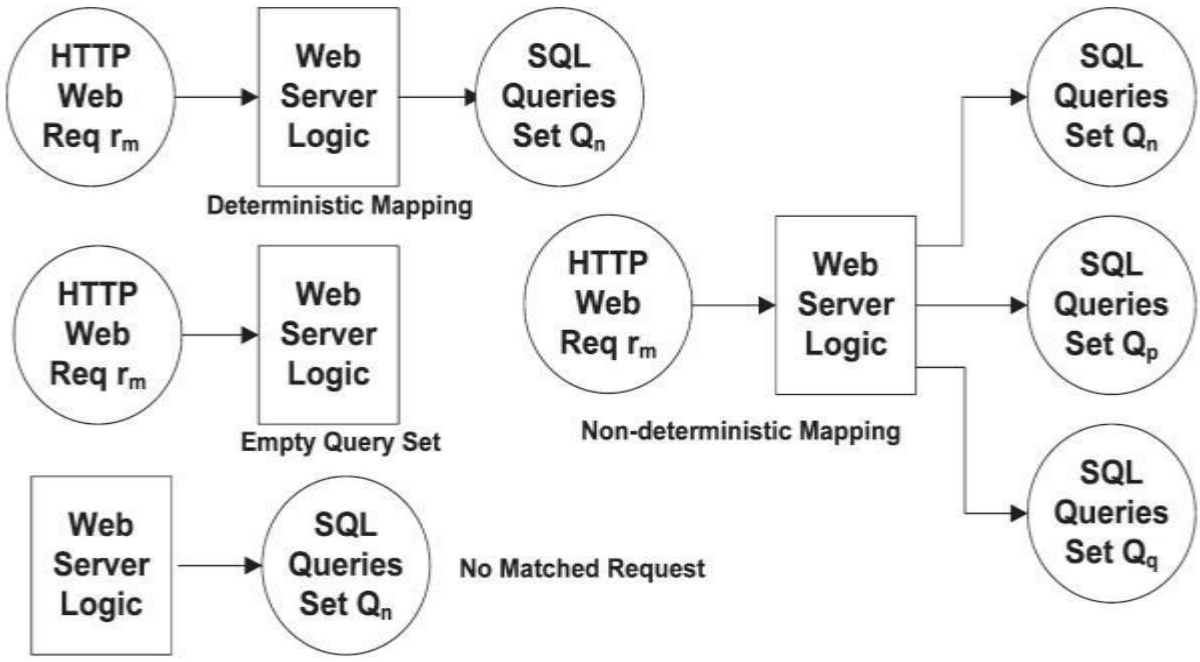


**Figure.6 Representation of Mapping Patterns**

**Detecting Intrusions**

After the normality model has been generated it can be worked out for training and detection of the abnormalities. Each session is then compared to normality model. We measure the web request with the mapping rule. Each discrete web request in the session will consist of only one mapping rule in the model. It can be done using the following algorithm:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

**Intrusion Detection Algorithm:**

1. In the deterministic mapping rule, where r ->Q, we check whether the query Q is a subset of the query set of the session. If it is there, then we mark those queries in Q as a valid request.

2. If the EQS i.e. r->ø, rule persists then the request is not being considered to be abnormal and we do not require to mark any database queries. So, no intrusion is being reported.

3. If the left out queries are unmarked, then there is a need to look out whether they are in set NMR. If it is so then they are marked as such. The queries which are new or unmarked are considered to be abnormal. If it exists within a session then that session will be marked as vulnerable.

## V. CONCLUSION AND FUTURE WORK

The multitier web analyser is developed to model the behaviour of the web applications. To detect the anomalous behaviour of the multitier web applications both at the front end as well as the back end data, the mapping model is used. The container-based architecture provides diverse session ID's for different HTTP requests which is helpful in isolating the information flow of all web server sessions. Thus, the multitier web analyser is able to recognize wide range of attacks invading the system. We achieved this by isolating the flow of information from each web server session with a lightweight virtualization. Furthermore, we quantified the detection accuracy of our approach when we attempted to model static and dynamic web requests with the back-end file system and database queries. In future work we detect and prevent sql injection attack and Privilege Escalation Attack by using causal mapping between web request and database queries.