Front End and Back End Database Protection from Intruders in Distributed Web Application

Shyam A Gade¹, Prof. M. B. Vaidya²
¹Student, ME Comp. A.V.C.O.E. Sangamner.
²Asst.Prof. A.V.C.O.E. Sangamner.

Abstract—Study of the intrusion detection system. Current network System has large scale trouble from the security vulnerability due to simple routing protocol, security less application, bugs in network & operating system. Therefore need of network intrusion detection system arrives. Generally there are two type of intrusion detection technique. First is misuse of computer resource detection & anomaly detection.

In misuse detection pattern there is some information available regarding well-known attack, which is used to identify the attack by comparing it with every incoming request. But this technique not in use full future or unrecognized attack. Which is not store or available in database of IDS.

Second method of anomaly detection used to technique to find if accessing of database is from normal usages method or not, which are based on some algorithm. Which are more sophisticated. Second method is most demand now a day than previous signature base detection.

Current day services moves to multitier design of web in which web server started in front end logic & data is kept in database server or file server.

Current paper we has present dual protection for both the front end logic & back end logic or file server by monitoring web request & database request.

Keywords: - Sql Injection, Privilege Escalation Attack, Future Session Attack, Multitier web Application, Direct Database attack.

I. INTRODUCTION

Now a day web application becomes the efficient common way to prepare services & make data available on network. As complexity of application increases vulnerability of application to intrusion attack also increases. Daily task such as banking, social friend chatting, visiting place are done through web.

These services uses web which is server front end logic which runs on user interface and back end server which consist of database or file server.

Due to use of web server personally or corporately it is obvious that web server may be getting attacked by intruders. Previous these attack were make on web server but now a day these attack become more diverse, attack attention shifted from web server to back end database system.

E.g. SQL injection attack [8][19]. However there is very little work being performed on multitier anomaly detection. In multitier architecture back end database usually protected by firewall while web server can be access remotely. Through this protect from direct remote attack but back end system may be come in under attack.

Intrusion detection system is used for detecting known threat or attack by comparing the traffic pattern [3-6] IDS and database both separately can be detected abnormal traffic send to either to them. But if attacker use normal traffic pattern to attack web server and database server these IDS cannot detect it. E.g. If any attacker who has not privileges of administrator log on web server by used user access process, he or she can be issue database query to exploited vulnerability of web server. In this case both web IDS and database IDS cannot detect the attack. It is because web IDS think that it is typically user login traffic while database server IDS will think that it is normal traffic of privileged user. This kind of attack easily detected if database server IDS will recognize a privileged request from web server is not associated with user privileged access. But bad luck it is not possible in current multi-threaded web server.

It is not possible to profile such causal mapping between web server traffic and database server traffic, since it is not attributed to user session.

In this paper we present dual protection to web server and database server. Systems which will be detect the attack in multitier web services. Our model differential user session which will web front end (HTTP) and back end SQL file. For this we implement technique to assign each user session to a dedicated container, which is a virtual computing environment. We will give container ID to each web request with its database queries. In this way by dual protection we will make causal mapping profile by considering both web server and database traffic.

II. RELATED WORK

As we have seen network intrusion detection system has two types:
Anomaly detection.

2. Misuse or behavior detection.

In anomaly detection intrusion detection system first decide what is correct and which state should accept in static from and dynamic behavior of the system. IDS use this result for detect abnormal changes or anomalous activity.

Behavior or misuse model are built by storing previous history of attack occurred [16], [23], [11]. An anomaly detector then compares actual usage pattern implemented established model to discover that event which are not normal.

Intrusion alert correlation [22] which tells us combination of different part which transforms IDS alerts into intrusion report so that reduced replicated alert negative positive alert. This paper also tells us one attack describing different level of alert. It concentrating on abstracting low level sensor attack and provide logical higher level compound alert to user.

But in our proposed dual protection we will feed multiple traffic to a single intrusion detection system in session so that it will produced about without correlating alert produced by other independent intrusion detection system.

An intrusion detection system such as [18] uses the temporary event to detect intrusion but in our dual protection does not correlated event. In dual protection is on time basis, because of the risk of mistakenly considering event but concurrent event as correlated event. In dual protection this type of event will handle by container ID to each session to casually map the related event. There is no problem that they are concurrent or not.

The database should receive highest level of protection because it contains more valuable information, so that more valuable research efforts have been made on database intrusion system. 14, 13, 20 and database firewall [9].

Some type of software like green SQL [7] work reverse proxy to database connection. Web server not connected database server directly instead of, they first connects database firewall first, and where SQL query are analyzed for safety if it safe then and then they are given to database server.

In some previous approaches of intrusion or vulnerabilities detection are done [1] by analyzing the source code or executable [52, 10, 12]. In some other approaches it dynamically track information and detect intrusion [2] [17,21,24].

CLAMP [15] is architecture for avoiding data leakage even in presence of attack. It isolates the code at the web server and data at the database layer from user. CLAMP gives guarantees user security data.

In contrast double guard [7] focuses on modeling mapping pattern between HTTP request and database queries for detection of unwanted user. In CLAMP existing application code is modified and all database access goes through the queries restrictor, which play role as a proxy mediator.

1) System Architecture:

In this approach we will use dual protection like above figure for protection of web server and database server. Here each direct request is assigned new session which is isolated and consist of both request i.e. HTTP request and back end request (SQL request). Each session assign new protector (we can also call it as container). We will give each protector separate no so that it recognize differently from other session or other protector.

In case of static website if web services can permit the back end data modification which also called as dynamic web services, they allow modification of HTTP request to include parameter which is not fixed and depend upon input given by user. So that ability of the model causal relationship between web servers is not always deterministic and depend upon application logic e.g. database queries are very based on value given password in HTTP request and previous application state.

But in some cases application main functionality like accessing table can be triggered by many various web pages. Therefore web and database requests resulting mapping can range from one to many depending upon value which are passed in parameter in the web request.
A. Proposed Work:

Break down structure of our approach will focuses of following areas:

1. User Control.
2. Session Monitoring.
3. Query Monitoring.
4. Mapping HTTP Queries with SQL Queries.
5. Showing Attack Log.

1) User control:-

**Input:** - User will do registration with getting user name and password.

**Output:** - user will login successfully or unsuccessful

**Algorithm Step:**

1. New user will fill registration page.
2. He will get login name with password.
3. User will log into system.
4. He will status his session.
5. After finishing work user log out.

Above algorithm describe how security is provided to the entire system so that it will prevent unauthorized access. If any new user wants to enter in the system he must be filled the registration form first. Here he must provide personal information with login name and password after sharing; this information will saved in database.

Now this user has its user name and password. After user entering list user name and password in login page successful message will be display if given login name and password are correct otherwise invalid username or password will be displayed. In this way user control provide security.

2) Session Monitoring:-

**Input:** - HTTP query and SQL query

**Output:** - provide Session ID to each request ‘r’ and SQL query ‘q’.

**Algorithm Step:**

1. For each session traffic T do.
2. Get different HTTP request ‘r’ and database query ‘q’ in this session.
3. For each various ‘r’ do.
4. If ‘r’ is a request to static file then.
5. Add ‘r’ into set EQS (empty query set)
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 SQL then
12. Add ‘q’ into SQL
13. Append session ID ‘i’ to the set AQq with ‘q’ as the key.

This module is responsible for giving unique identification number to HTTP request and SQL request. If HTTP request is present in web server then ‘r’ is added to empty query set this query will not get any identification numbers. If ‘r’ is not in set of REQ i.e. query is now of arrives in first time in web server then it is added to REQ.

3) Query Monitoring:-

**Input:** - HTTP query ‘r’ and SQL query ‘q’.

**Output:** - insertion of queries into query set.

**Algorithm Step:**

1. For each session separated traffic T do
2. Get different HTTP request ‘r’ and database query ‘q’ in this session.
3. For each ‘r’ do
4. If ‘r’ is a request to static file then
5. else.
6. Add ‘r’ into set EQS (empty query set)
7. If ‘r’ is not set in REQ then
8. Add ‘r’ into REQ
9. For each different ‘q’ do
10. If ‘q’ is not set SQL then
11. Add ‘q’ into SQL
Query monitoring is the module in which different query request are added in query set. If any query is present in data set or file then ‘r’ is added EQS (empty query set). If ‘r’ is not present in query set means it is new and arrives first time then it is added into Req (REQUEST QUERY SET).

Likewise each SQL query if ‘q’ is not present into SQL query then it is added into SQL set.

SQL query then it is added into SQL set.

4) Mapping HTTP Queries with SQL Queries:-

Input:- ARr and AQq are set and t is cardinality

Output:- HTTP query mapped to equivalent SQL query.

Algorithm Step:-

1. For each distinct HTTP request ‘r’ in Req do
2. For each DB query ‘q’ in SQL do
3. Compare set of ARr with set of AQq
4. If ARr = AQq then return false
5. If ARr = AQq and cardinality (ARr >t) then
6. Found a deterministic mapping from ‘r’ to ‘q’
7. Add ‘q’ into mapping model set MSr or ‘r’
8. Mark ‘q’ in set SQL
9. Else
10. Need more training session
11. Return false
12. For each DB query ‘q’ in SQL do
13. If ‘q’ is not marked then
14. Add ‘q’ into set NMR (No Matched Request)
15. For each HTTP request in Req do
16. If ‘r’ has no deterministic mapping model then
17. Add ‘r’ into set EQS (Empty Query Set)
18. Return True

User send request to web server in the form of HTTP. Then web server generates the equivalent SQL query. This query mapping (monitoring) map HTTP query with SQL query. This module used the session monitoring module and query monitoring module output.

5) Attack Detection:-

Input:- HTTP query ‘r’ and SQL query ‘q’.

Output:- log which will show attack

Algorithm Step:-

1. If the rule for request is deterministic mapping r -> Q (P ≠φ) we will test whether Q is a subset of query, if it is subset of query then it is valid, and we will mark queries in Q. otherwise we will considered something going to be wrong and session will considered as suspicious
2. If rule is empty query set r-> φ then request should be normal but not do any database queries and no any attack will be reported.
3. For remaining unmarked queries we will check whether these database queries are present in No Match Request (NMR) pattern.
4. Any unchecked web request database queries should be abnormal consider. If these are present in session then it should be considered suspicious.

6) Handling of Dynamic Web Page:-

Dynamic web pages are different than static web pages. Dynamic web page will give permission to generate web queries which are same to different parameter.

A dynamic web page generally uses POST method instead of GET method, to commit user’s inputs.

Base on web server application logic different input causes different database queries. E.g. for post a comment on blog article, web server would send the query to database first to see existing comment.

If user has send different request from previous request then web server automatically generate new queries and store it into database otherwise web server will reject that query so that no duplicate query will be stored. In this case assigning same parameter value generates different set of queries which is depending upon previous state of web site.

This non deterministic mapping case like one to many mapping happen even after normalizing all parameter values to extract structure of web request of web queries. The mapping appears differently in each case so it is not easy to recognize all one to many mapping pattern in each web request. It is also possible if different operation occasionally overlap at different their possible query set, it become very difficult to us to extract one to many mapping for every operation by comparing web request and queries in across the session.

Algorithm which is used in static pages for extraction of mapping pages is not useful in dynamics web pages, so we can create different method. We can create category for each operation and store these operation in single category which are somewhat alike. E.g. common operation on user blog may be reading or posting article or giving any kind of comment or visiting to next page etc. all these operation are in same session.

If we able to design mapping model for such kind of basic operation and to design set of queries, mapped from these basic operation and if any single operation could not contains that kind of basic request and queries in a particular session then this may be an intrusion.
This can be done as each request rm or set of related request Rm in various session with various input we can prepare many queries sets like Qn, Qp, Qq.

Then we can prepare such operation model as Rm -> Qm (Qm= Qn v Qp V Qq…………) in this relation Rm is set of web request and Qm are queries which may be triggered by that operation. One thing must be kept in mind that both deterministic and non-deterministic mapping and set of Empty Query Set (EQS) are still used to hold static file request.

B. Intrusion Detection in Dynamic Website:-

As if we build separate mode for single operation, we can use it to detection of intrusion by observing abnormal session. In final phase of testing we can compare traffic captured in each session with model. We also examine all operation that this request is belongs. Since one request may be present in many another models we can take set of such entire query from CQS (Collection Query Set).

E.g. in session ‘i’ the set of data base queries Qi should sub set CQS. If we found some unmatched queries in session, this will indicate that session has violated the mapping model. For each web request Ri each query should be matched at least one request of model or be in the set of EQS.

For example we can take model which has two single operations such as reading and writing article. Consider mappings are as below read-> RQ and write -> WQ.

If we test then in session ‘i’ all request in the session we will see that it is from any group it is either read or either write i.e. there two basis operation in one session. Therefore query set Qi must be such that it is subset of RQ U WQ which is Collection of Query Set (CQS). If any query found in this session which is not belong to either of basic operation then it may be harmful or intrusion likewise if we found web request in session which is not in any kind of model which we have covered previously may be an abnormal web request by our algorithm we can found such kind of suspicious request in our detection.

III. CASE STUDY OF ATTACKS

Direct Database Attack:-

Some times of traffic or request that is not go through any container webservice or firewall but connect directly to the database. Database queries will not having any matching web request during this type of attack. A web server’s intrusion detection system could not detect this kind of attack.

If any attacker launched this kind of attack this can be easily identified by our approach, since we cannot match any web request with these queries.

SQL injection Attack:-

SQL injection attack is such type of attack in which it does not require to compromise the web server. Attacker only uses the existing pitfall or vulnerability to web server so that he can inject the data. When web server uses that data to use of backend the attacker obviously get what he wants to do.
As we are going to use double protection for web server and database server even if the exploited are accepted by web server, content of database server would not be able to take on expected structure for the given web server request.

**Hijack Future Session Attack:**

A session hijacking attack can be categories in various names like spoofing a denial of service attack or packet drop attack or man in the middle attack. These types of attacks are mainly effective on web server side.

In this case attacker first takes control over web server and hijack all authorized session for attack purpose. Sometime attacker can send spoofed reply or drop user request as showing this figure.

In abnormal situation no any conventional web server IDS nor a database IDS can detect such kind of attack.

Our container based web server can prevent such kind of attack because in our container based web server each user get new session in new container and attacker can never break other user session.

**Privilege Escalation Attack:**

Consider any web server give service to both regular user and administrator consider for regular user request may be Ru and will trigger SQL query Qu and for administrator request may be Ra and will trigger SQL query Qa.

Now consider attacker log into web server as a normal user and upgrade his privilege as an administrator and trigger a query for getting administrator data. These kinds of attack are very difficult to detect for web server intrusion detection system or database intrusion detection system. As both Ru and Qa are authorized request and authorized queries.

But by our container based approach we can find this kind of attack since database queries Qa will does not match request Ru.

**IV. Conclusion**

Due to high speed backbone and local area network give the user high band width connectivity. The internet becomes the vital infrastructure for comparing government and millions of user. But new technologies of computer network also become more complicated in web application. It is not possible to fulfil all security demand from traditional intrusion detection system.

We are going to present an intrusion detection system that will build model of normal behaviour of multtier distributed web application of front end web server and backend database queries previous approaches which has correlated alert generated by individual intrusion detection system but dual protection which will allowed multiple input requests to produce alert.

We will active this by isolating the information flow or request from each web server session with lightweight virtualization. We can also increase our accuracy of intrusion detection by attempting model of static and dynamic web request.

**REFERENCES**


