MIGRATING TOWARDS DOUBLE GUARD : CONTAINER BASED APPROACH TO DETECT INTRUSION IN WEB APPLICATION

Sayyad Rijwanali[1] , Kiran Joshi[2], Sowmiya Raksha[3]

[1]Department Of Computer Engineering , V.J.T.I. Mumbai (India)

[2,3]Assistant Professor , Department Of Computer Engineering & IT , V.J.T.I. Mumbai (India)

ABSTRACT

Intrusion detection systems are used to detect attacks against computer systems, networks. It is difficult to provide provably secure information systems and to maintain them in such a secure state during their lifetime. Network Intrusion Detection Systems (NIDS) face challenges coming from the network link speed and complexity of threats.

Internet plays a crucial role in our everyday life affects the individual life of every person in decisive ways .In this architecture web services have upgraded into a multitier architecture in which web server runs the application front-end logic and backend information is submitted to database server. Over past few years internet services and applications had raised the complexity of the attacker to attack the web server. So in order to avoid web attacks we should use container based approach to detect intrusion in web application. Our focus is to study Double Guard intrusion detection system. Double Guard provides a secure environment for multitier web application. Using double guard we can monitor the Web request and its subsequent database requests so we can find out attack that cannot possible with normal intrusion detection system.

Keywords : Double Guard, Anomaly Detection, Misuse Detection, Web Server, Database Server, Intruder.

I INTRODUCTION

Now-a-days, web-delivered services like banking, travel, social networking, shopping etc. become immensely very popular as well as extremely complicated. These services significantly uses a web- server at front front-end which runs the application program logic and a back-end database server that consists of a data or other information . Because of their regular use for confidential, personal information web dependent services have continuously been target for attacks. Because of shifting of attention from front end web server exploiting vulnerabilities of the online applications so as to corrupt the back-end information system through SQL injection [2]. Intrusion-detection systems aim to catch attacks against computer systems and networks, or against data systems normally, it is tough to produce incontrovertibly secure information systems and maintain them in such a secure state for their entire life time and for each utilization. To protect multitier web services, intrusion

detection systems are widely used to find attacks by matching misused pattern or signatures to protect multitier internet services. A category of intrusion detection system that uses machine learning can even finds unknown attacks by identifying abnormal network traffic from previous behaviour of intrusion detection system.

II BACKGROUND

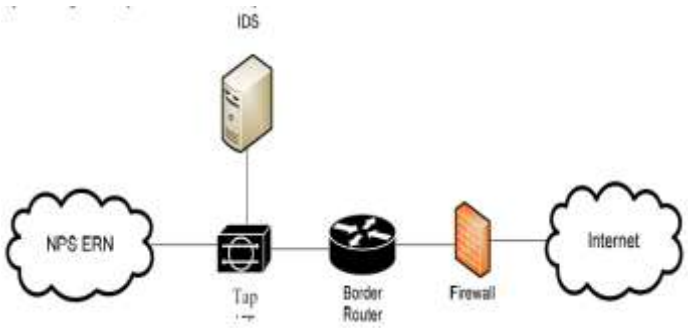

Fig. 1: NIDS Architecture

Intrusion detection is network-based when the system is used to analyze network packets. Network based Intrusion Detection System (NIDS) capture the network traffic from the wire as it travels to a host. This can be analyzed for a particular signature or for unusual or abnormal behaviours.

A Network Intrusion Detection System can be classified into two types

1.1 Anomaly Detection

In Anomaly detection, the proper and acceptable static and dynamic behaviour of the System is outlined and characterized first. This could be used to find out the changes or abnormal behaviours. Then an anomaly detector compares actual usage patterns against models that are already established so as to spot abnormal events. We tend to follow the anomaly detection approach since we rely on a training phase to make the correct model.

1.2 Misuse Detection

Misuse detection approach is used to detect attacks. In misuse detection approach, we have a tendency to outline abnormal system behaviour initially, so define the any other behaviour, as normal behaviour. It stands against anomaly detection approach that utilizes the reverse approach, process defining system behaviour and processes any other behaviour as abnormal.

II RELATED WORK

Both data and pipeline parallelism have been used in the context of NIDS. In pipeline parallelism, the whole packet processing is divided into several sequential stages that each run on a dedicated execution unit. A packet is transferred sequentially from one execution unit in the pipeline to the next. In addition to the parallelism gained, pipelining improves reference locality and potentially increases the cache hit ratio since each execution unit only deals with a subset of the entire application memory. NIDS even though it is parallel it cannot detect the web attacks.

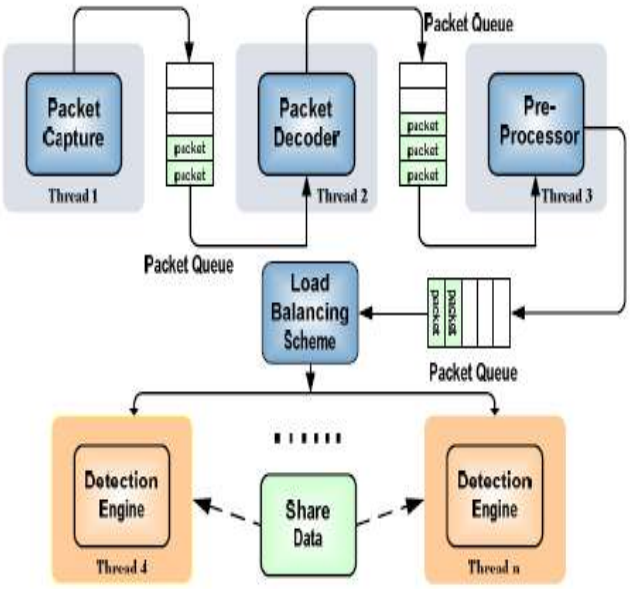

Fig. 2: Pipeline Based IDS Architecture

As web and also the database servers are vulnerable. Attacks are network borne and are available from the web clients; they will launch application layer attacks to compromise the web servers they're connecting to. The attackers will bypass the web server to directly attack the database server. we assume that the attacks will neither be detected nor prevented by this web server IDS, that attacker could take over the web server after the attack, which later on they will acquire full control of the web server to launch future attacks. As an example, the attackers could modify the application logic of the web applications, listen or hijack different users' web requests, or intercept and modify the database queries to steal sensitive beyond their privileges.

On the opposite hand, at the database end, database server won't be completely taken by the attackers. Attackers might strike the database server through the web server or, more directly, by submitting SQL queries, they'll get and pollute sensitive data inside the database. These assumptions are reasonable since, in most cases, the database server isn't exposed to the general public and thus tough for attackers to fully take over.

In this paper, we are considering the pitfalls of IDS, NIDS as they are not enough to protect against web based attacks. Here, we are migrating towards, Double Guard, a system which will be able to find out attacks in multitier web services. Double guard uses a light-weight virtualization technique to assign every user's web session to a dedicated container, which provides virtual computing environment and uses the container ID to accurately associate the web request with the subsequent database queries. Thus, Double Guard will build an effective mapping model by taking each the web server and database server traffic into consideration.

III DOUBLEGUARD SYSTEM ARCHITECTURE

In our architecture, we make use of light-weight process, mentioned as "containers," as passing, disposable servers for client sessions. It's possible to initialize thousands of containers on one physical machine, and these virtualized containers will be discarded, reverted, or quickly reinitialized to serve new sessions.

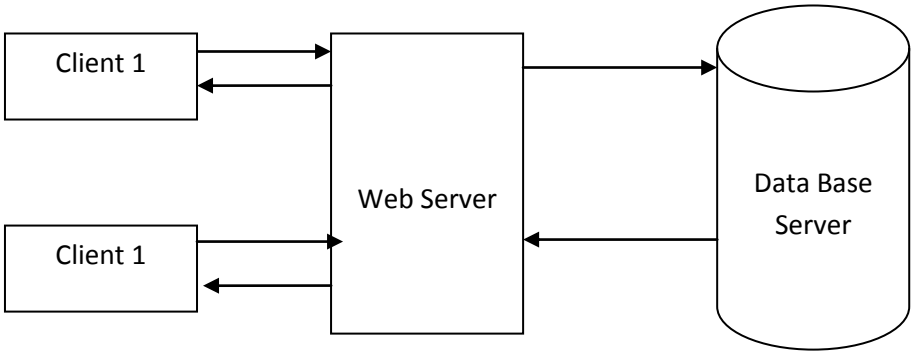

Fig. 3: Traditional three tier model. The web server used in the front end, and the database servers are used in backend.

This container-based, session-separated web server architecture enhances the security performances as well as provides us with the isolated information flows that are separated in each container for each session. It allows us to identify the mapping between the HTTP requests and the subsequent Database queries, and utilize such a mapping model to find out abnormal behaviour in session.

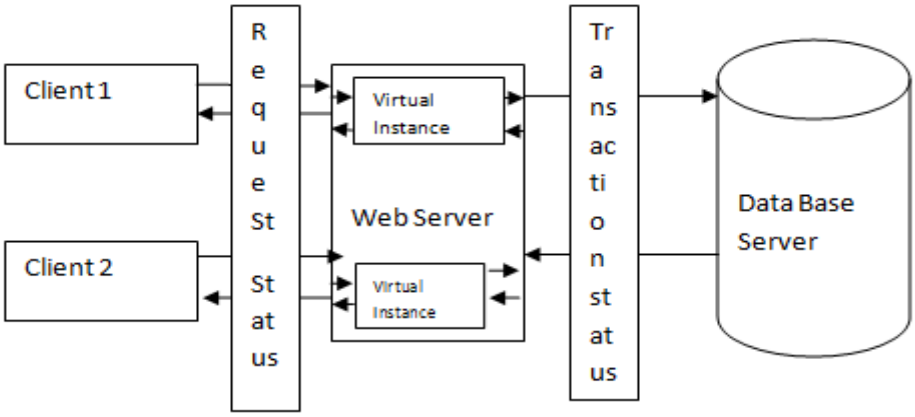

Fig. 4: Proposed architecture of double guard.

Once we learn the model, it can be used to find out abnormal behaviours. Both the HTTP request and the DB queries within each session should be matched with the model. If there exists any request or query that does not follow the normality model for session, then the session will be treated as a possible attack.

IV ATTACKS

4.1 Privilege Escalation Attack

In this attacker logs into the web server as a normal user, upgrades his/her privileges, and triggers admin queries thus he obtains an administrator's data. This attack will never be detected by either the web server IDS or the

database IDS. Our approach, however, will notice this kind of attack since the database query doesn't match the request, according to our mapping model.

4.2 Hijack Future Session Attack

This category of attacks is especially aimed at the web server side. An attacker typically takes over the web server and thus hijacks all succeeding legitimate user sessions to launch attacks.

According to the mapping model, the web request should invoke some database queries, and then the abnormal scenario is detected. However, neither a traditional web server IDS nor a database IDS will find such an attack by itself.

4.3 Injection Attack

Attacks like SQL injection don't need compromising the web server. Attackers will use existing vulnerabilities within the web server logic to inject the data or string content that contains the exploits and then use the web server to relay these exploits to attack the back end database. Since our approach provides a two-tier detection, even if the exploits are accepted by the web server, the relayed contents to the database server wouldn't be able to take on the expected structure for the given web server request.

4.4 Direct DB Attack

It is potential for an attacker to bypass the web server or firewalls and connect on to the database. An attacker may even 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 might detect neither. Moreover, if these database queries were within the set of allowed queries, then the database IDS wouldn't detect it either. However, this sort of attack will be caught with our approach since we cannot match any web requests with these queries.

V ALGORITHM FOR MAPPING WEB REQUEST WITH DATABASE QUERIES

Input: Training Data set, Threshold t

1. for each session separated traffic Ti do
2. Get different web requests r and database queries q in this session
3. for each different web requests r do
4. if r is a request to static file then
5. Add r into set EQS(Empty Query Set)
6. else
7. if r is not in set REQ then
8. Add r into REQ
9. Append web 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 web session ID i to the set AQq with q as the Key
14. for each distinct web request r in REQ do [8]

5.1 Advantages of Intrusion Detection System

1. Accuracy

The accuracy of Intrusion Detection System is to detect attacks that are based on pair types and signatures. To observe such attacks in multitier web applications an IDS uses internet IDS and database IDS [5].

2. Performance

The performance of an intrusion detection system is that the rate at which audit events are processed. If the performance of the intrusion-detection system is good, then it's possible to observe real-time attack [5].

3. Timeliness

An intrusion-detection system performs and propagates its analysis as quickly as potential so that the security officer able to react before abundant harm has been done, and also to prevent the attacker from subverting the audit source or the intrusion-detection system itself [5].

Limitations of Intrusion Detection System

Drawbacks of Intrusion Detection System include the problem of gathering the required data on the known attacks and keeping it with new vulnerabilities and environments. Also in Intrusion Detection System an attacker will directly attack backend database server [5].

VI CONCLUSION

In this paper, we are considering the pitfalls of IDS, NIDS as they are not enough to protect against web based attacks. Here, we are migrating towards Double guard IDS.

We discussed an Intrusion Detection System that builds models of traditional behaviour for multitier applications from each front-end web (HTTP) requests and back-end database (SQL) queries. It forms container based IDS with multiple input streams to provide alerts.

This builds models of traditional behaviour for multi tiered web applications that prevent each a network IDS like Snort or a database IDS from detecting attacks against these vulnerabilities. However, by observing the mapping relationship between web requests and database queries, Double Guard is effective at capturing such attacks. We presented an Intrusion Detection System that builds models of traditional behaviour for multitier applications from each front-end web (HTTP) requests and back-end database (SQL) queries. It forms container based IDS with multiple input streams to provide alerts. We've seen that such correlation of input streams provides a much better characterization of the system for anomaly detection. So, migrating towards double guard IDS is better than other IDS.

REFERENCES

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

[2] Niraj Gaikwad 1, Swapnil Kandage 2, Dhanashri Gho lap," Double Guard: Detecting & Preventing Intrusions in Multitier web applications", http://warse.org/pdfs/2013/ ijns02222013.

[3] "Common Vulnerabilities and Exposures," http://www.cve. mitre.org/, 2011.Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

[4] William Robertson, "Effective Anomaly Detection with Scarce Training Data". https://www.cs.ucsb.edu/2010.

[5] Greensql, http://www.greensql.net/, 2011.

[6] 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.

[7] H. Debar, M. Dacier, and A. Wespi, "Towards Taxonomy of Intrusion-Detection Systems," Computer Networks, vol. 31, no. 9, pp. 805-822, 1999.

[8] Gopale Sheetal S 1, Gamane Sonali S. 2 , Monica Bachal K. 3,"DoubleGuard: Intrusion Detection System in Web Application"

Biographical Notes

Sayyad Rijwanali is currently pursuing M. Tech final year in Computer Engineering Department (Specialization in Network Infrastructure Management System) from V.J.T.I, Mumbai, India.

Kiran Joshi is working as Assistant Professor in Computer Engineering & IT Department, V.J.T.I Mumbai, India.

Sowmiya Raksha is working as Assistant Professor in Computer Engineering & IT Department, V.J.T.I Mumbai, India.