ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Complete Monitoring Of Web by Session Control Using Multi-Tier Container Concept

Vincila.A[1], Gopalakrishna Moorthy.K[2], Saranya.R[3]

PG Student, Dept. of CSE, Parisutham Institute of Technology and Science, Thanjavur, Tamilnadu, India[1]

Assistant Professor, Dept. of CSE, Parisutham Institute of Technology and Science, Thanjavur, TamilNadu, India[2]

PG Student, Dept. of CSE ,Parisutham Institute of Technology and Science,Thanjavur,TamilNadu,India[3]

Abstract -Today web applications have become a ubiquitous use of daily tasks such as banking, travel and social networking; etc. Web services provide communication and management of personal information from anywhere. With the increase in the use of these applications, there is an increase in the number and complexity of vulnerabilities and attacks. Due to their ubiquitous use for everyday tasks, web applications have always been the target of attacks. Intrusion Detection System (IDS) is proposed with lightweight virtualization that models the network behaviour for web applications. By monitoring the web and subsequent database requests, it can identify the attacks and provide early warning about suspicious activities. Casual mapping model is developed between web server requests and subsequent database queries which identifies various kinds of attacks. Each session is assigned with each container for single user. The container ID accurately associates the web request with the subsequent DB queries. Session control is used to control each session for each user request. Honey pot method is implemented to collect the intruder's behaviors and stores that details in session block list table. Whenever a new session is tracked for Intrusion Detection, it is first referred with the session block list. If the tracked session is matched with the session block list table then it is avoided to enter into the session. Hence, the Intrusion avoidance can be achieved in an efficient manner.

Keywords - Intrusion detection system, Mapping model, Virtualization, Container, Session block list table, Honey pot.

I. INTRODUCTION

Web-Delivered services and applications have increased in both popularity and complexity over the past few years. Daily tasks like banking, travel, and social networking, are all done through the web. Web services typically employ a webserver front end that runs the application user interface logic method, as well as a back-end server that consists of a database or file server. Due to their global use for personal or corporate data, attackers are trying to attack the web services.

A plethora of Intrusion Detection Systems (IDSs) currently examine network packets individually within both the webserver and the database system. However, there is very little work performed on multitier Anomaly Detection (AD) systems that generate models of network behavior for both web and database network interactions. In that kind of multitier architectures, the back-end database server is protected behind a firewall while the webservers are remotely accessible over the Internet. Unfortunately, even though they are protected from direct remote attacks, the back-end systems are vulnerable to attacks that use web requests as a means to exploit the back end.

To protect the multitier web services, IDS have been widely used to detect known attacks by matching misused traffic patterns or signatures. A class of IDS that forces machine learning can also detect unknown attacks by identifying abnormal network traffic that deviates from the so-called "normal" behavior previously profiled during the IDS training phase. Separately, the web IDS and the database IDS can detect abnormal network traffic sent to either of

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

them. However, it found that these IDSs cannot detect cases wherein normal traffic is used to attack the webserver and the database server. For example, if an attacker who is not having admin privileges can log in to a webserver using normal-user access credentials, attacker can find a way to issue a privileged database query by exploiting vulnerabilities in the webserver.

Neither the web IDS nor the database IDS would detect this type of attack since the web IDS would merely see typical user login traffic and the database IDS would see the normal traffic of a privileged user. This type of attack can be detected if the database IDS can identify that a privileged request from the webserver is not associated with user-privileged access. Unluckily, within the current multithreaded webserver architecture, it is not practical to detect or profile such causal mapping between webserver traffic and DB server traffic since traffic cannot be clearly attributed to user sessions.

Intrusion Detection system is used to detect attacks in multitier web services. This approaches can create normality models of isolated user sessions that include both the web front-end (HTTP) and back-end (File or SQL) network transactions

II. RELATED WORK

A network Intrusion Detection System can be classified into two types: anomaly detection and misuse detection. Anomaly detection requires the IDS to define and characterize the correct and acceptable static form and dynamic behavior of the system, which can be used to detect abnormal changes or behaviours. The boundary between acceptable and anomalous forms of stored code and data is precisely definable. Normal behaviour models are built by performing a statistical analysis on historical data or by using rule-based approaches to specify behaviour patterns.

An anomaly detector then compares actual usage patterns against established models to identify abnormal events. This detection approach belongs to anomaly detection, and it depends on a training phase to build the correct model. Since some legitimate updates may cause model drift, there are a number of approaches that are trying to solve this problem. The detection may run into the same problem; in such a case, 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 and non-relevant positives. It fuses the alerts from different levels describing a single attack, with the goal of producing a brief overview of security-related activity on the network. It first focuses on abstracting the low-level sensor alerts and providing logical, high-level alert events to the users.

III. PROPOSED SYSTEM

The Intrusion avoidance can be obtained by implementing the session block list Table. The Intruders identified using the proposed work can be blocked. This blocking can be done with honey pot method. Honey pot is a network server to trap attacker before they invade the real systems. It provides the environment where intruders can be trapped. Then it collects intruder's information such as ip address, logging files and their records.

The detected intruders' details can be collected and stored in the session block list. Whenever a new session is tracked for Intrusion Detection, it is first referred with the session block list. If the tracked session is in the block list then it is avoided to enter into the session. Hence, the Intrusion Avoidance can be achieved in an efficient manner.

The hidden attacks can be identified using the IDS. This approach can effectively identify the various types of attacks such as injection attack, direct DB attack, session hijacking, privilege escalation attack, Cross site scripting attack and DDOS attack.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

HONEY POT

Honey pot systems are decoy servers or systems setup to gather information regarding an attacker into the system. Honey pot is built by setting up the server with the tempting files. Make it hard but not impossible to break into system. Observe them as they cavort around in the server. Log their conversations with each other and study their activities.

TRAFFIC COLLECTION & CONTAINER CREATION

Web traffic is the volume of data sent and received by visitors to a website, which is determined by the number of visitors and the number of pages they visit. Websites monitor the incoming and outgoing traffic to see which parts or pages of their site are popular and if there are any seeming trends, such as one specific page being viewed mostly by people in a particular country.

CONTAINER CREATION

A web application runs within a Web container of a Web server. Web container provides the runtime environment through components that provide naming context and life cycle management. The basic idea of Servlet container is using Java to dynamically generate the web page on the server side. Hence servlet container is essentially a part of a web server that interacts with the servlets.

In this, each user session is assigned into a different container. For instance, a new container per each new IP address of the client is assigned. In this implementation, containers were recycled based on events or when sessions time out. In this design, lightweight process containers are used which are referred to as “containers,” as transient, disposable servers for client sessions.

It is possible to initialize thousands of containers on single physical web machine, and these virtualized containers can be discarded or quickly reinitialized to serve new sessions. As a result, a single physical server can run continuously and serve all web requests. On the other hand, from a logical perspective, each session is allotted to a dedicated webserver and isolated from other sessions.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

ARCHITECTURE DIAGRAM

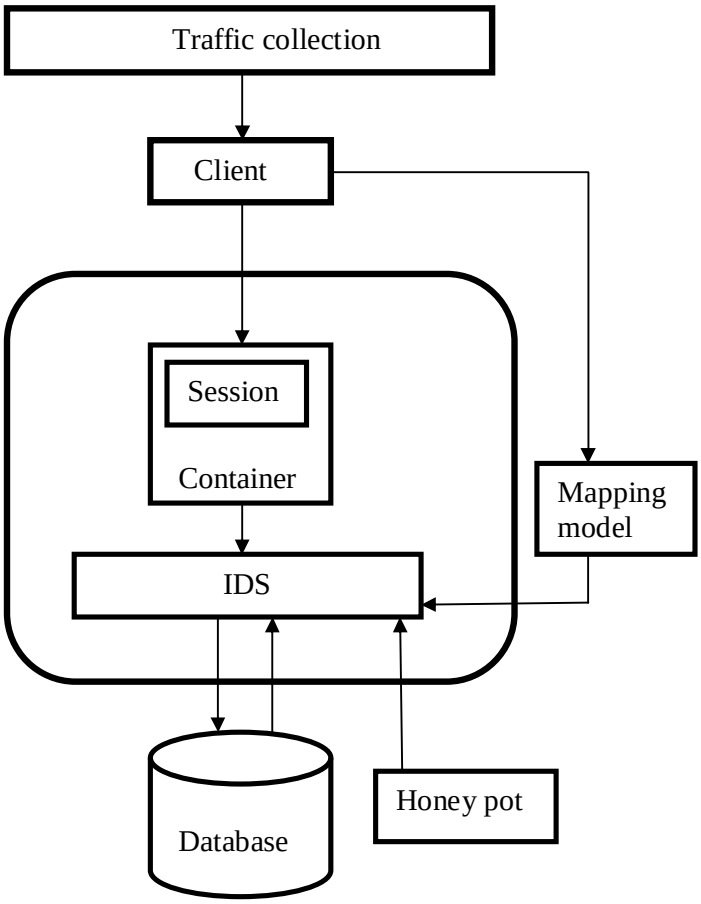

Fig 1: Architecture diagram

The available users are collected by traffic collection. Each session is allotted for each user and that is put into the container. Container is created by light weight virtualization technique. So numbers of containers are created on single physical machine. Mapping model method is used to match the webserver request with the subsequent database queries.

Intrusion detection system monitors the individual packet in the network to avoid the intrusions. It gets current user's information from mapping model and container. Honey pot is implemented by providing the environment with the tempting files. It is setup to be easier prey for intruders than true production systems but with the minor system modifications. So that their activity can be monitored. It can have the intruder's subsequent visits. During subsequent visits additional information can be gathered.

Honey pot stores traced intruder's activities into the session block list table. This session block list table is connected with the IDS. Whenever a new session is tracked first that is referred with the session block list table. If traced session

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

is matched with the table, it is avoided to enter into the session. Then it blocks that user. Otherwise it allows current user as legitimate user.

Modelling deterministic mapping and patterns

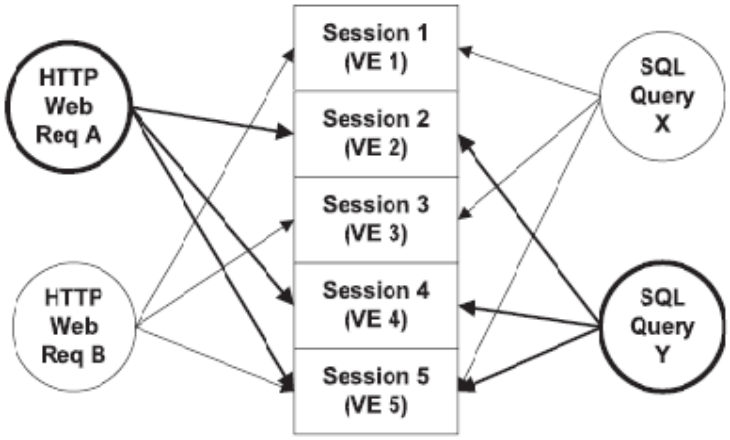

Fig 2: Deterministic mapping using session ID of the container

Deterministic mapping

This is the most common and perfectly matched pattern. The web request rm appears in all traffic with the SQL queries set Qn. For any session in the testing phase with the request rm, the absence of a query set Qn matching the request indicates a possible intrusion. On the other hand, if Qn is present in the session traffic without the corresponding rm, this may also be the sign of an intrusion. In static websites, this type of mapping comprises the majority of cases since the same results should be returned for each time a user visits the same link.

Empty query set

In special cases, the SQL query set may be the empty set. This implies that the web request neither causes nor generates any database queries. For example, when a web request for retrieving an image GIF file from the same webserver is made, a mapping relationship does not exist because only the web requests are observed. During the testing phase, it keeps these web requests together in the set EQS.

No matched request

In some cases, the webserver may periodically submit queries to the database server in order to conduct some scheduled tasks, such as cron jobs for archiving or backup. This is not driven by any web request, similar to the reverse case of the Empty Query Set mapping pattern. These queries cannot match up with any web requests, and it keeps these unmatched queries in a set NMR. During the testing phase, any query within set NMR is considered legitimate. The size of NMR depends on webserver logic, but it is typically small.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Nondeterministic mapping

The same web request may result in different SQL query sets based on input parameters or the status of the webpage at the time the web request is received. In fact, these different SQL query sets do not appear randomly, and there exists a candidate pool of query. Each time that the same type of web request arrives, it always matches up with one (and only one) of the query sets in the pool.

TABLE 1: Detection Results for Attacks

Operation	Snort	GSQL	DG
Privilege Escalation (WordPress Vul)	No	No	Yes
Web Server aimed attack (nikto)	Yes	No	Yes
SQL Injection (sqlmap)	No	Yes	Yes
DirectDB	No	No	Yes
linux/http/ddwrt_cgibin_exec∗	No	No	Yes
linux/http/linksys_apply_cgi∗	No	No	Yes
linux/http/piranha_passwd_exec∗	No	No	Yes
unix/webapp/oracle_vm_agent_utl∗	No	No	Yes
unix/webapp/php_include∗	Yes	No	Yes
unix/webapp/php_wordpress_lastpost∗	No	No	Yes
windows/http/altn_webadmin∗	No	No	Yes
windows/http/apache_modjk_overflow ∗	No	No	Yes
windows/http/oracle9i_xdb_pass∗	No	No	Yes
windows/http/maxdb_webdbm_database∗	No	No	Yes

IV. CONCLUSION

In this an intrusion detection system, intruders are identified and blocked using session block list table which has the collection of intruder's activities. Whenever a new session is tracked for Intrusion Detection, it is first referred with the session block list. If the tracked session is in the block list then it is avoided to enter into the session. Hence, the Intrusion Avoidance can be achieved in an efficient manner. Using this method, easily it can identify the hidden attacks. IDS build models of normal behaviour for multitier web applications is presented from both front-end web (HTTP) requests and back-end database (SQL) queries. Unlike previous approaches that correlated or summarized alerts generated by independent IDSs, it forms container-based IDS with multiple input streams to produce alerts. This is achieved by isolating the flow of information from each webserver session with a lightweight virtualization.

REFERENCES

[1] Anley .C, "Advanced Sql Injection in Sql Server Applications, "technical report, Next Generation Security Software,Ltd., 2002.
[2] Bai .K, Wang .H and Liu .P, "Towards Database Firewalls," Proc.Ann. IFIP WG 11.3 Working Conf. Data and Applications Security (DBSec '05), 2005.
[3] Barry .B.I.A and Chan H.A, "Syntax, and Semantics-Based Signature Database for Hybrid Intrusion Detection Systems," Security and Comm. Networks, vol. 2, no. 6, pp. 457-475, 2009.
[4] Bates .D, Barth .A, and Jackson .C, "Regular Expressions Considered Harmful in Client-Side XSS Filters," Proc. 19th Int'l Conf. World Wide Web, 2010.
[5] Christodorescu .M and Jha .S, "Static Analysis of Executable to Detect Malicious Patterns," Proc. Conf. USENIX Security Symp.,2003.
[6] Kim H.A and Karp .B, "Autograph: Toward Automated Distributed Worm Signature Detection," Proc. USENIX Security Symp., 2004.
[7] Kruegel .C and Vigna .G, "Anomaly Detection of Web-Based Attacks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), Oct. 2003.
[8] Debar .H, Dacier .M, and Wespi .A, "Towards Taxonomy of Intrusion-Detection Systems," Computer Networks, vol. 31, no. 9, pp. 805-822, 1999.