ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Network Intrusion Detection and Prevention Using K-Means Clustering

Bhor Priyanka M[1]
Department of Computer Engineering,
Jaihind Collage of Engineering, Kuran,
Pune, India

Rajguru Punam S[2]
Department of Computer Engineering,
Jaihind Collage of Engineering, Kuran,
Pune, India

Asware Vaishali A[3]
Department of Computer Engineering,
Jaihind Collage of Engineering, Kuran,
Pune, India

Prof. N. N. Shaikh[4]
Department of Computer Engineering,
Jaihind Collage of Engineering, Kuran,
Pune, India

Abstract: ***Internet one of the important means of communication. Internet services and applications have become an inextricable part of daily life, enabling communication and the management of personal information from anywhere. Web services have advanced to multi-tiered design wherein the web server runs various applications at front-end logic and data are outsourced to file server or database. Computer networks are nowadays subject to an increasing number of attacks. By monitoring both web and consequent database requests, we are able to ferret out attacks that independent IDS would not be able to identify. Intrusion Detection Systems (IDS) are designed to protect them by identifying malicious behaviors or improper uses. Intrusion detection is the process of monitoring and analyzing the events occurring in a computer system in order to detect signs of security problems. Today most of the intrusion detection approaches focuses on the issues of feature selection or reduction. Since some of the features are irrelevant and redundant they result to lengthy detection process and degrades the performance of an intrusion detection system (IDS). In this paper, we present Double Guard, a system used to detect attacks in multi-tiered web services. . Our 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. The purpose of this study is to identify important reduced input features in building IDS that is computationally efficient and effective. The detection accuracy when system attempt to model static and dynamic web requests with the backend file system and database queries is quantified by Double Guard system. We quantify the shortcomings of any multitier IDS in terms of training sessions and functionality coverage Using this we will be able to expose various types of attack like Denial Of Service Attack (DOS), SQL Injection.***

Keywords: Double Guard, Database Server, Client, Intruder, Web Server, K-Means Clustering.

I. INTRODUCTION

The Internet is a global system of interconnected computer networks that make use of standard protocol suites. The Internet is world widespread network. Security is a ultimate component of every network design. Hackers are skilled programmers who understand the details of every network, computer communications and have knowledge of how to exploit the vulnerabilities. Web based attacks have recently become more diverse, as attention has shifted from attacking the front-end to exploiting vulnerabilities of the web applications in order to corrupt the back-end database system (e.g., SQL injection attacks). A plethora of Intrusion Detection Systems (IDS) currently examine 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 generate models of network behavior for both web and database network interactions. In such multi-tiered architectures, the back-end database server is often protected 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 means to exploit the back-end. 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. A class of IDS that leverages 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. Individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them. However found that these IDS cannot detect cases wherein normal traffic is used to attack the web server and the database server. For example, if an attacker with non admin privileges can log in to a web server using normal-user access credentials, he/she can find a way to issue a privileged database query by exploiting vulnerabilities in the web server. 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 only the normal traffic of a privileged user. This type of attack can be readily detected if the database IDS can identify that a privileged request from the web server is not associated with user-privileged access. Unfortunately, within the current multi-threaded web server architecture, it is not feasible to detect or profile such causal mapping between web server traffic and DB server traffic since traffic cannot be clearly attributed to user sessions.

II. Literature Survey

Intrusion Detection System (IDS) Intrusion Detection Systems have three main components Network Intrusion Detection system (NIDS), Network Node Intrusion detection system (NNIDS) and Host Intrusion Detection Sys-tem (HIDS). Intrusion detection systems have been widely used to detect known attacks by matching misused traffic patterns or signatures [3]. Intrusion Detection Systems is the security management technique that leverages machine learning which detects unknown attacks by identifying abnormal network traffic that deviate the so-called “normal” behavior. An intrusion is defined as set of procedures that compromises of integrity, confidentiality or availability of a resource [1]. Intrusion detection is the process of monitoring and analyzing the suspicious events occurring in a computer system in order to detect signs of security problems. There are two main strategies of IDS: misuse detection and anomaly detection. Misuse detection attempt to match signatures and patterns of already known attack in the network traffic. A constantly updated database is usually used to store the signatures of known attacks. Anomaly detection attempts to identify behavior that does not correspond to normal behavior. This technique is based on the detection of traffic anomalies. The anomaly detection systems are adaptive in nature, they can deal with new attacks, but they cannot identify the specific type of attack. Many researchers have proposed and implemented various models for IDS but they often generate too many false alerts due to their simplistic analysis [3].

III. Existing System

Intrusion detection system examines the network packets within both the web server and the database system individually. However, there is very little work being performed on multi-tiered Anomaly Detection system that produces models of network behavior for both web and database interactions. Web services and database servers are vulnerable. Attacks are generally done by the web clients. The attackers try to bypass web servers to attack the database server directly. Database servers are often protected by the firewall while the web servers are remotely accessible over the internet. Unfortunately, even after protecting them from direct remote attacks, the back-end systems are susceptible to attacks that use web request as a means to exploit the back end. Web IDS would merely see typical user login traffic and database IDS see normal traffic of privileged user. Attackers could modify the application logic of the web applications, eavesdrop or hijack another user’s web requests, or intercept and modify the database queries to steal sensitive data beyond their privileges.

IV. PROPOSED SYSTEM

In Double guard detection using both front end and back end detection. Some previous approaches have detected intrusions or vulnerabilities by statically analyzing the source code or executable others dynamically track the information flow to understand taint propagations and detect intrusions. In double guard, the new container based web server architecture enables us to separate the different information flows by each sessions by using light weight virtualization. Within a light weight virtualization environment we ran many copies of web server instances in different containers so that each one isolated from the rest.it separate different information flow from the each session. This provides a means of tracking the information flow from the web server to the database server for each session. It is possible to initialize the thousands of container on a single machine.

Double guard detects sql injection attacks by taking the structure of the web request and database queries without looking into the values of input parameter. In our Double Guard, we utilize the container ID to separate session traffic as a way of extracting and identifying causal relationship between web server request and database query event. Our approach dynamically generates new containers and recycles used ones. As a result a single physical server can run continuously and serve all web requests. However, from a logical perspective, each session is assigned to a dedicated web server and isolated from other sessions. Since initialize each virtualized container using a read only clean template, can guarantee that e a c h session will be served with a clean web server instance at initialization.

This system chooses to separate communications at the session level so that a single user always deals with the same web server. Sessions can represent different users to some extent, and we expect the communication of a single user to go to the same dedicated web server, thereby allowing us to identify suspect behavior by both session and user. If detect abnormal behavior in a session, will treat all traffic within this session as tainted.

The system architecture of Double guard is represented in Figure 1 as shown below. In Double guard architecture, we are using the lightweight virtualization technique to assign a separate web container to each user. Each user will have a separate web container ID for processing web request. The use of container makes it easy for initializing, destroying and lasting for only short-time, which provides a single container for each user. It is possible to initialize thousands of containers on a single system and these virtualized containers can be discarded, reverted or quickly reinitialized to serve new sessions. In Double guard approach, new container and recycle used container are dynamically generated [2]. Figure 1 illustrates the architecture of Double guard. The client generates an HTTP request and sends it to the Database server. The Server on the other hand receives the request and then processes this request. The Server also plays an important role by managing sessions, maintaining connection with the database and examining attacks. Each session is assigned to a dedicated web server and separated from other sessions. The server produces the result by generating and processing the queries. This result is then sent as a response to the client's request. The server generates the log files. Meanwhile, if we found or detect abnormal behavior in a session, we will treat all the network traffic within the session as infected.

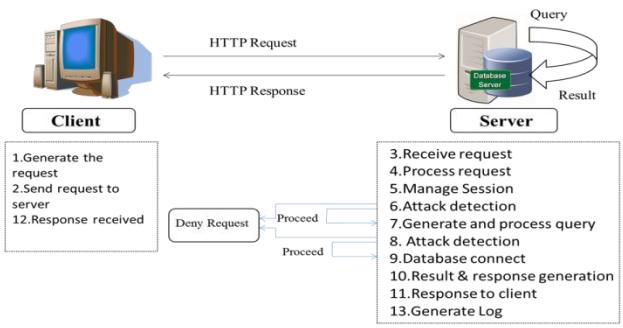


Figure 1: System architecture of Double Guard

V. K-MEANS ALGORITHM

The K-means algorithm, starting with k arbitrary cluster centers in space, partitions the set of giving objects into k subsets based on a distance metric. The centers of clusters are iteratively updated based on the optimization of an objective function. This method is one of the most popular clustering techniques, which are used widely, since it is easy to be implemented very efficiently with linear time complexity. The principle goal of employing the K-Means clustering scheme is to separate the collection of normal and attack data that behave similarly into several partitions which is known as K^{th} cluster centroids. In other words, K-Means estimates a fixed number of K, the best cluster centroid representing data with similar behavior. In our work, we predefined K=2, representing Cluster 1, Cluster 2. Thus, the iterative K-Means algorithm is designed as follows:

Initially: Randomly select K = 2 cluster centroid .Do Correspond data point to nearest clusters. Update optimal cluster centroid based on corresponding data points and labeling the points while no change remains certain activities or data are alike to either normal or abnormal behavior. The k-Means algorithm groups N data points into k disjoint clusters, where k is a predefined parameter.

Detection with k-Means Clustering:

Client sends multiple requests to the server

Let $X = \{x_1,x_2,x_3,\ldots\ldots..,x_n\}$ be the set of data points and $V = \{v_1,v_2,\ldots\ldots.,v_c\}$ be the set of centers.

1) Randomly select *'c'* cluster centroid. (Initially usercentroid and robotcentroid as zero.)

2) Generate the two clusters i.e usercentroid and robotcentroid clusters then calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

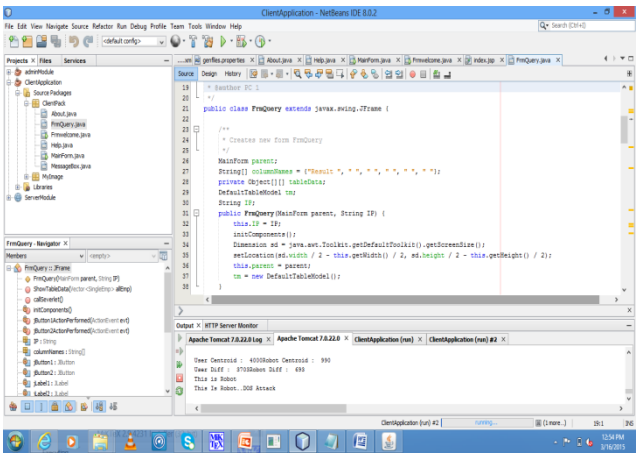
4) Calculate the request difference and detect the attack:

If userdifference < Robotdifference

Normal User is detected.

If userdifference > robotdifference

Robot attack is detected.



VI. ATTACK SCENARIOS

Our system is effective at capturing the following types of attacks:

1) SQL Injection attack

An SQL injection attack works by crafting SQL states which are combined with the contents submitted by Web pages. Techniques commonly used in SQL attacks include comment symbols, identical equations (such as 1 = 1), union queries by using the union statement, and inserting or modifying data by using the insert or update statement. SQL injection attacks are much more likely to happen than other Web attacks and cause more widespread harms. Such harms include obtaining the system control right, operating data-bases without authorization, tempering web page content and adding system accounts or data-base user accounts. As the numbers of attack targets and attackers increase, there have been more and more SQL injection attacks in recent years. Attackers can use existing vulnerabilities in the web server logic to inject the data content that contains the exploits and then use the web server to relay these exploits to attack the back-end database.

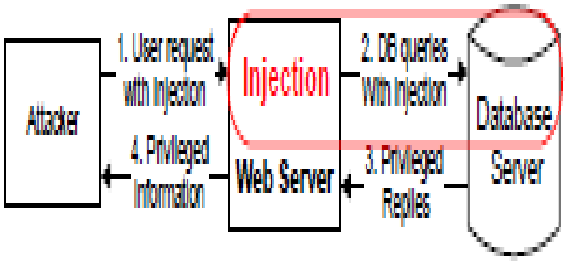


Figure 2: SQL Injection

2) DOS Attack

A denial-of-service (DoS) or distributed denial-of-service (DDoS) attack is an attempt to make a machine or network resource unavailable to its intended users. A DoS attack generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root name servers. Denial-of-service threats are also common in business and are sometimes responsible for website attacks.

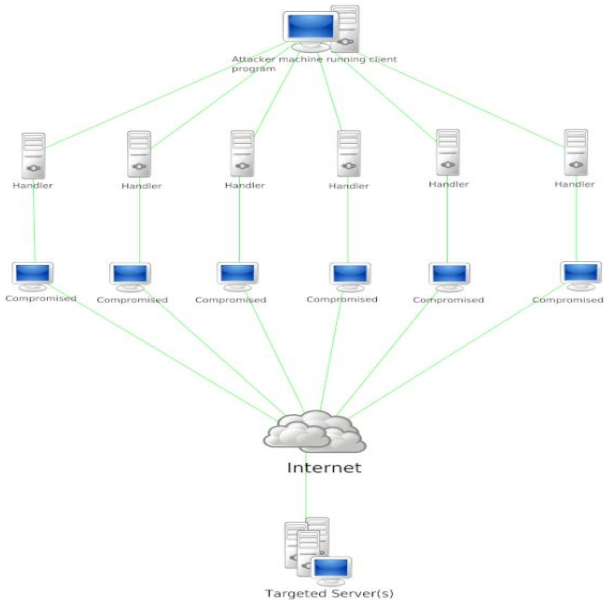


Figure 3: DOS Attack

VII. ADVANTAGES

*1) **Accuracy***

The accuracy of Intrusion Detection System is brilliant to detect attacks that are based on mismatch types and signatures. To detect such attacks in multitier web applications an IDS uses web IDS and database IDS.

*2) **Performance***

The performance of an intrusion detection system is the rate at which audit events are processed. If the performance of the intrusion-detection system is excellent, then it is possible to detect real-time attacks

*3) **Timeliness***

An intrusion-detection system performs and propagates its analysis as quickly as possible so that the security officer is able to detect the attacks and prevent the damage that is provoked due to these attacks. It also prevents the attacker from subverting the audit source or the intrusion-detection system itself.

VIII. CONCLUSION

Double guard presented an intrusion detection system that builds models of normal behavior for multi-tiered web applications 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, Double guard forms container-based IDS with multiple input streams to produce alerts. We have shown that such correlation of input streams provides a better characterization of the system for anomaly detection because the intrusion sensor has a more precise normality model that detects a wider range of threats. For static websites, we built a well-correlated model, which our experiments proved to be effective at detecting different types of attacks.

References

1. SANS, “The Top Cyber Security Risks,” http://www. sans. org/ topcyber-security-risks/, 2011.
2. Meixing Le, Angelos Stavrou, Brent ByungHoon Kang “Doubleguard: Detecting Intrusions in Multitier Web Applications,” IEEE Transactions On Dependable and Secure Computing , vol. 9,No. 4, July/August 2012
3. “Network Intrusion Detection System (NIDS) Using Data Mining Techniques” [Online] Available on http://etrx. spit. ac. in/ieee_colloquium/Information_Security/spit-265. pdf.
4. “A Data Mining Framework for Building Intrusion Detection Models1”. [Online] Available on, http://citeseerx.ist.psu.edu/viewdoc/download.
5. “Common Vulnerabilities and Exposures,” http://www. cve. mitre. org/, 2011.
6. 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
7. Five common web application vulnerabilities. http://www. symantec. com/connect/articles/five-common-web-application-vulnerabilities.
8. T. Verwoerd and R. Hunt. “Intrusion detection techniques and approaches”. Computer Communications, 25(15), 2002.
9. Gerhard M¨unz, Sa Li, Georg Carle “Traffic Anomaly Detection Using K-Means Clustering”. Computer Science.
10. Mrs. Ghatge Dipali D.” Network Traffic Intrusion Detection System using Decision Tree & K-Means Clustering Algorithm”, Volume 2, Issue 5, September – October 2013.
11. Vivek Vashishtha, Durgesh kumar, “ IDS Improved with K-Means Algorithms, Self Organizing Map and Auto Class”, Computer Science and Software Engineering, Volume 2, Issue 5, May 2012.
12. Harshit Saxena, Dr. Vineet Richariya,” Intrusion Detection System using K- means, PSO with SVM Classifier: A Survey”, Computer Science Engineering, Volume 4, Issue 2, February 2014.
13. Farhad Soleimanian Gharehchopogh, Neda Jabbari, Zeinab Ghaffari Azar, “ Evaluation of Fuzzy K-Means And K-Means Clustering Algorithms In Intrusion Detection Systems”, VOLUME 1, ISSUE 11, DECEMBER 2012.
14. A. M. Riad, Ibrahim Elhenawy, Ahmed Hassan, Nancy Awadallah, “VISUALIZE NETWORK ANOMALY DETECTION BY USING K-MEANS CLUSTERING ALGORITHM”, Computer Science and Information Systems, Vol.5, No.5, September 2013.

