ISSN 2395-1621

New Publicly Verifiable Databases Supporting Intrusion Detection System

[#1]Aniruddha Teke, [#2]Tanmay Yadav, [#3]Hrishikesh Navale, [#4]Pratik Phatangare,
[#5]Prof. Ismail Mohammed

[2]yadavtanmay97@gmail.com
[5]issu4uever@gmail.com

[#12345]Department of Computer Science & Engineering,

Sinhgad Academy of Engineering, Kondhwa-411048, Pune, India*

ABSTRACT

An intrusion detection system that builds models of normal behavior for multitiered 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 a container-based IDS with multiple input streams to produce alerts. To improve mechanism to detect intrusions in multitier web applications Http attack system uses lightweight process containers referred to as "containers," as ephemeral, disposable servers for client sessions. It is possible to initialize thousands of containers on a single physical machine, and these virtualized containers can be discarded, reverted, or quickly reinitialized to serve new sessions.

ARTICLE INFO

Article History

Received: 9th April 2018

Received in revised form :

9th April 2018

Accepted: 11th April 2018

Published online :

11th April 2018

Keywords- Double Guard, Intrusion Detection System (IDS).

I. INTRODUCTION

This project considers a variety of application level threats facing enterprise web applications and how those can be mitigated in order to promote security. Evidence shows that perhaps as many as sixty percent of attacks on enterprise web applications are facilitated by exploitable vulnerabilities present in the source code. In this paper we take the approach of examining the various threats specific to the application layer along with their corresponding compensating controls. Threats specific to each of the tiers of the n-tiered enterprise web application are discussed with focus on threat modelling. Compensating controls are addressed at the architecture, design, implementation and deployment levels.

A. Objective

We Propose an efficient IDS system called as Http Attack system that models the network behavior for multilayered web applications of user sessions across both front-end web (HTTP) request and back-end database (SQL) quires.

B. Overview

The scope of our project is to design and implement our system by using software and hardware. The ultimate goal is that the ideas and planning demonstrated through this model system can then be easily upgraded to an actual e-commerce website. As the system is to be implemented on large shopping websites, there are a number of performance specifications that have to be met to ensure the system operates correctly and efficiently. Most importantly, The Development of our system interface must send and receive the appropriate information. The interface in turn, must be able to multi-task and have numerous threads running at the same time in order to track multiple users throughout the system functions. Presently we are focusing on small dataset of one website and minimum user's data.Most large organizations have hundreds of servers with thousands of directories and files stored on them and specific types of data that needs to be tagged. The software can be useful for identifying well-defined content like Social Security or credit cards numbers) but tends to fall short when an administrator is trying to identify other sensitive data, like intellectual property that might include graphic components, formulas or schematics.

II. LITERATURE SURVEY

Title	Author	Year	Review
The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information	Suphannee Sivakorn, *Iasonas Polakis and Angelos D. Keromytis*	2016	Our study revealed numerous instances of major services exposing private information and protected account functionality to non-authenticated cookies. This threat is not restricted to websites, as users' cookies are also exposed by official browser extensions, search bars and mobile apps.
Ontology based Intrusion Detection System for Web Application Security	Mr. Harshal A. Karande, Assist Prof. Shyam S. Gupta	2015	Attack ontology can be used intelligently for earlier attack detection. Attack ontology is a powerful input to intrusion detection system. Requests can be parsed, validated and forwarded for extracting the data.
A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks	Piyush A. Sonewar , Nalini A. Mhetre	2015	We have identified the threats of SQL injection and XSS attack. The proposed model is developed using .net framework of Windows operating system. Additional security measures can be provided using stored procedures. This approach applies mapping model to detect SQL injection and XSS attacks.
DoubleGuard: Detecting Intrusions in Multitier Web Applications	Meixing Le, Angelos Stavrou	2016	An intrusion detection system that builds models of normal behavior for multitiered 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, DoubleGuard forms a container-based IDS with multiple input streams to produce alerts.

III. SYSTEM ARCHITECTURE AND DESIGN

In this section, we present the architecture and design of proposed reservation based smart parking system, which implements a reservation service to systematically park the vehicles and earn money .

A. Proposed System

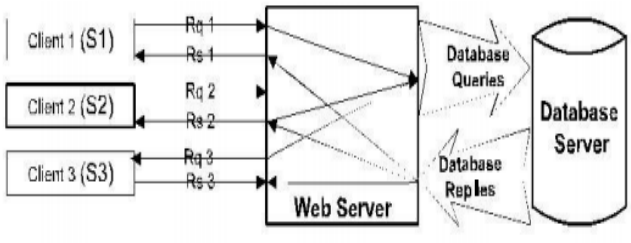

Fig.1 Architectural diagram of Http attack

We initially set up our threat model to include our assumptions and the types of attacks we are aiming to protect against. We assume that both the web and the database servers are vulnerable. Attacks are network borne and come from the web clients; they can launch application-layer attacks to compromise the web servers they are connecting to. The attackers can bypass the web server to directly attack the database server. We assume that the attacks can neither be detected nor prevented by the current web server IDS, that attackers may take over the web server after the attack, and that afterward they can obtain full control of the web server to launch subsequent attacks. For example, the attackers could modify the application logic of the web applications, eavesdrop or hijack other users‟ web requests, or intercept and modify the database queries to steal sensitive data beyond their privileges. On the other hand, at the database end, we assume that the database server will not be completely taken over by the attackers. Attackers may strike the database server through the web server or, more directly, by submitting SQL queries, they may obtain and pollute sensitive data within the database. These assumptions are reasonable since, in most cases, the database server is not exposed to the public and is therefore difficult for attackers to completely take over. We assume no prior knowledge of the source code or the application logic of web services deployed on the web server. In addition, we are analysing only network traffic that reaches the web server and database. We assume that no attack would occur during the training phase and model building.

To improve mechanism to detect intrusions in multitier web applications Http attack system uses lightweight process containers referred to as "containers," as ephemeral, disposable servers for client sessions. It is possible to initialize thousands of containers on a single physical machine, and these virtualized containers can be discarded, reverted, or quickly reinitialized to serve new sessions. In the classic three-tier model database side, it is unable to tell which transaction corresponds to which client request. The

communication between the web server and the database server is not separated, and we can hardly understand the relationships among them.

B. System Feature

Our system mainly focuses on providing safe and secure environment to all users while they are doing online payment or online shopping.
There are various attacks that can be introduced in this online virtual world. As we are giving ease and comfort to users but along with this various attacks are also getting added because we are giving our personal information in online network.

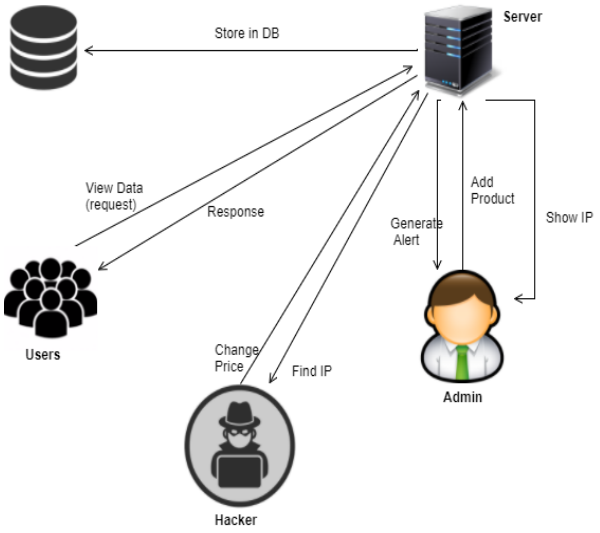

Fig.2 System Feature

Any third party user trying to do attack on system can easily get all users personal and bank information.
So main feature of system is to detect those attacks and prevent those attacks. This provides safety and assurance to users that our important data is safe on network.

C. Double Guard Algorithm

1. HTTP request from client identified to know either it's a query or an appeal.
2. Input value stored in hash table with appropriate key values for category AQ for query and AR for request.
3. Check the Query, if empty query then set as abnormal else proceed.
4. Condition with key provided to Request(AR) and query(AQ).
5. If Condition acknowledged then it terminates the HTTP request automatically by the virtual system.
6. Else send data or information requested by client
7. Exit

To protect multitier web applications against attacks is the main concentration point in Intrusion Detection System using DoubleGuard. We propose a prototype of DoubleGuard using a web server with a back-end DB. We also propose some modification to existing DoubleGuard to increase its performance, reliability in case of static and dynamic web sites. In our prototype, we propose to assign each user session into a different container; however, this will be a design decision.

D. Data Leakage Algorithm

1. User registers to system
2. User performs login by entering user_id and password If (authenticated) then Send web request Else Invalid User
3. Assign separate access as per user privileges
4. Check web request If request correct or wrong add those to the log list
5. Web server generates OTP
6. Evaluate query If (unauthorized) then Generate alert and reject Else Forward query to database server
7. Perform query processing
8. Output

With the fast growth of database business on the net, the data may be unsafe after passing through the unsecure network. The data purchasers may hesitate to buy the data service for the following suspicion. First, the data receiver may suspect that the data are tampered with by unauthorized person. Second, they may suspect the data received are not produced and provided by the authorized suppliers. Third, the suppliers and purchasers actually with different interest should have different roles of rights in the database management or using. So how to protect and verify the data becomes very important here. The recent surge in the growth of the internet results in offering of a wide range of web-based services, such as database as a service, digital repositories and libraries, e-commerce, online decision support system etc. In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties.

For example, a hospital may give patient records to researchers who will devise new treatments. We call the owner of the data the distributor and the supposedly trusted third parties the agents. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data.

IV. Conclusion

In this paper, an intrusion detection system that builds models of normal behavior for multitiered 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, DoubleGuard forms a container-based IDS with multiple input streams to produce alerts. Attack ontology can be used intelligently for earlier attack detection. Attack ontology is a powerful input to intrusion detection system. Requests can be parsed, validated and forwarded for extracting the data.

REFERENCES

[1] Suphannee Sivakorn, Iasonas Polakis and Angelos D. Keromytis: "The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information",2016.

[2] Mr. Harshal A. Karande, Assist Prof. Shyam S. Gupta: "Ontology based Intrusion Detection System for Web Application Security",2015.

[3] Piyush A. Sonewar , Nalini A. Mhetre: " A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks",2015.

[4] Meixing Le, Angelos Stavrou.: "A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks", 2016.

