ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 1, January 2013

# VirtuaGuard: Intrusion Detection System on Static and Dynamic Web Applications

AJINKYA NIKAM BHIM BIRADAR SAGAR DHERE PRAHSANT SONAWANE

Prof .Ruta Kulkarni

*Zeal Education Society's Dnyanganga College of Engineering and Research, Pune 411041*

***Abstract*—**In this project, we propose an efficient IDS system called as VirtuaGuard system that models the network behavior for multilayered web applications of user sessions across both front-end web (HTTP) requests and back-end database (SQL) queries. In this system, VirtuaGuard forms container-based IDS with multiple input streams to produce alerts. In typical three-tiered web server architecture, the web server receives HTTP requests from user clients and then issues SQL queries to the database server to retrieve and update data. This proposed container-based and session-separated web server architecture enhances the security performances and also provides the isolation between the information flows that are separated in each container session. In order to detect the abnormal behaviors on a session/client level, Casual Mapping profile model is newly developed to map between the web server requests and the subsequent DB queries.

## 1. INTRODUCTION

Web applications are the most common way to make services and data available on the Internet. Unfortunately, with the increase in the number and complexity of these applications, there has also been an increase in the number and complexity of vulnerabilities. Services and applications have become an inextricable part of daily life, Internet enabling communication and the management of personal information from anywhere. To accommodate this increase in application and data complexity, web services have moved to a multi-tiered design wherein the web server runs the application front-end logic and data are outsourced to a database or file server.

VirtuaGuard differs from this type of approach that correlates alerts from independent IDSs. Rather, VirtuaGuard operates on multiple feeds of network traffic using single IDS that looks across sessions to produce an alert without correlating or summarizing the alerts produced by other independent IDSs. This 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. For websites that do not permit content modification from users, there is a direct causal relationship between the requests received by the front-end webserver and those generated for the database back end. No prior knowledge of the source code or the application logic of web services deployed on the webserver.

Virtualization is used to isolate objects and enhance security performance. Lightweight containers can have considerable performance advantages over full virtualization

We present VirtuaGuard, 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. To achieve this, we employ a lightweight virtualization technique to assign each user's web session to dedicated container, an isolated virtual computing environment. We use the container ID to accurately associate the web request with the subsequent DB queries. Thus, Virtua Guard can build a causal mapping profile by taking both the web server and DB traffic into account.

## 2. RELATED WORK

Intrusion detection is performed by analyzing one or more input event streams, looking for the manifestation of an attack. Historically, detection has been achieved by following one of two different approaches: anomaly detection or misuse detection. Anomaly detection relies on models of the "normal" behavior of a computer system. These models can focus on the users, the applications, or the network. Behavior profiles are built by performing a statistical analysis on historical data or by using rule-based approaches to specify behavior patterns. An anomaly detector then compares actual usage patterns against established profiles to identify abnormal patterns of activity.

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 positives, and non-relevant positives.

It also fuses the alerts from different levels describing a single attack, with the goal of producing a succinct overview of security-related activity on the network. It focuses primarily on abstracting the low-level sensor alerts and providing compound, logical, high-level alert events to the users. VirtuaGuard differs from this type of approach that correlates alerts from independent IDSs. Rather, VirtuaGuard operates on multiple feeds of network traffic using single IDS that looks across sessions to produce an alert without correlating or summarizing the alerts produced by other independent IDSs . Validating input is useful to detect or prevent SQL or Cross Site Scripting (XSS) injection attacks. This is orthogonal to the VirtuaGuard approach, which can utilize input validation as an additional defense. However, we have found that VirtuaGuard can detect SQL injection attacks by taking the structures of web requests and database queries without looking into the values of input parameters (i.e., no input validation at the webserver).Virtualization is used to isolate objects and enhance security performance.

Full virtualization and Para-virtualization are not the only approaches being taken. An altar-native is a lightweight virtualization, such as OpenVZ, Parallels Virtuozzo or Linux-V Server . In general, these are based on some sort of container concept. With containers, a group of processes still appears to have its own dedicated system, yet it is running in an isolated environment. On the other hand, lightweight

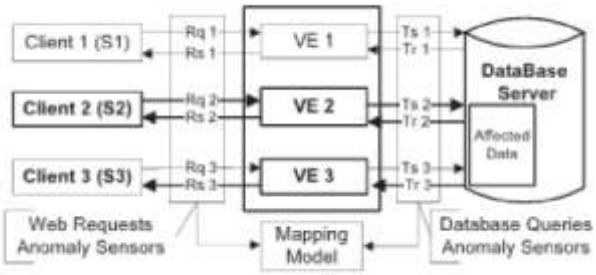
Containers can have considerable performance advantages over full virtualization or Para-virtualization. Thousands of containers can run on a single physical host. There are also some desktop systems that use lightweight virtualization to isolate different application instances. Such virtualization techniques are commonly used for isolation and containment of attacks. However, in our VirtuaGuard, we utilized the container ID to separate session traffic as a way of extracting and identifying causal relationships between webserver requests and database query events

## 3. THREAT MODEL AND SYSTEM ARCHITECTURE

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 webservers they are connecting to. The attackers can bypass the webserver to directly attack the database server. We assume that the attacks can neither be detected nor prevented by the current webserver IDS, that attacker may take over the webserver after the attack, and that afterward they can obtain full control of the webserver 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.

### 3.1 ARCHITECTURE AND CONFINEMENT

We make use of 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. A single physical webserver runs many containers, each one an exact copy of the original webserver.




### 3.2 ATTACKS SCENARIOS

1. Privilege Escalation Attack
2. Hijack Future Session Attack
3. Injection Attack
4. Direct DB Attack

## 4. MODELING FOR STATIC WEBSITES

The nondeterministic mapping does not exist as there are no available input variables or states for static content. We can easily classify the traffic collected by sensors into three patterns in order to build the mapping model. As the traffic is already separated by session, we begin by iterating all of the sessions from 1 to N. For each rm 2 REQ, we maintain a set ARm to record the IDs of sessions in which rm appears. The same holds for the database queries; we have a set AQs for each qs 2 SQL to record all the session IDs. To produce the training model ,we leverage the fact that the same mapping pattern appears many times across different sessions. For each ARm,we search for the AQs that equals the ARm. When ARm ¼ AQs, this indicates that every time rm appears in a session, then qs will also appear in the same session, and vice versa.

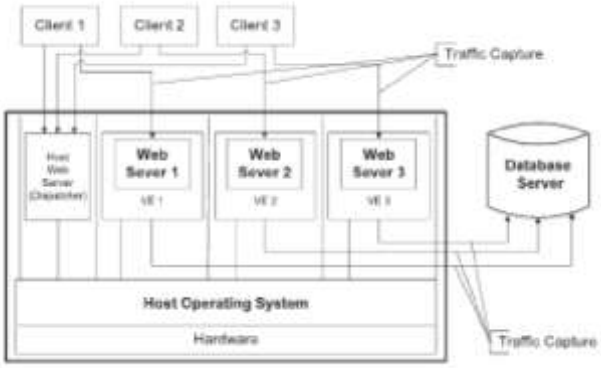
## 5. ATTACK DETECTION

We used the sql map, which is an automatic tool that can generate SQL injection attacks. Nikto , a webserver scanner tool that performs comprehensive tests, and Metasploit were used to generate a number of webserver-aimed http attacks (i.e., a hijack future session attack). We performed the same attacks on both VirtuaGuard and a classic three-tier architecture with a network IDS at the webserver side and a database IDS at the database side.

## 6. IMPLEMENTATION

We implemented a prototype of VirtuaGuard using a Webserver with a back-end DB. We also set up two testing websites, one static and the other dynamic. To evaluate the

detection results or our system, we analyzed four classes of attacks described in section 3.2 we chose to assign each user session into a different container; however, this was a design decision. For instance, we can assign a new container per each new IP address of the client. In our implementation, containers were recycled based on events or when sessions time out. We were able to use the same session tracking mechanisms as implemented by the Apache server (cookies ,mod_user track, etc.) because lightweight virtualization containers do not impose high memory and storage overhead.




**7. CONCLUSION:**

This paper presented an approach for multitier web applications intrusion detection, called VirtuaGuard. The approach is implemented by extending the general IDS performs detection of web-based attacks.

We have achieved this by isolating the flow of information from each webserver session with a lightweight virtualization. Another important aspects we quantified the detection accuracy of our approach when we attempted to model static and dynamic web requests with the back-end data system and database queries. Which our experiments proved to be effective at detecting different types of attacks. Moreover, we showed that this.

## 8. REFERENCES

1.SANS, "The Top Cyber Security Risks," http://www.sans.org/top-cyber-security-risks/, 2011.

2. National Vulnerability Database, "Vulnerability Summary for CVE-2010-4332," http://web.nvd.nist.gov/view/vuln/detail? vulnId= CVE-2010-4332, 2011.

3. National Vulnerability Database, "Vulnerability Summary for CVE-2010-4333," ttp://web.nvd.nist.gov/view/vuln/detail? vulnId=CVE-2010-4333, 2011.

4.Autobench, http://www.xenoclast.org/autobench/, 2011.

5."Common Vulnerabilities and Exposures," http://www.cve. mitre. org/, 2011.

6. "Five Common Web Application Vulnerabilities," http://www. symantec.com/connect/articles/five-common-web-application-vulnerabilities, 2011.

7. greensql, http://www.greensql.net/, 2011.