A Review on DoubleGuard: Detecting Intrusion in Multitier Web Application

Abhaya Raut\textsuperscript{1}, Aditi Shah\textsuperscript{2}, Chaitrali Pande\textsuperscript{3}, Prof. Arya C.S.\textsuperscript{4}
\textsuperscript{1,2,3} Student, BE Computer, JCOE, Department Of Computer Engineering, Kuran \textsuperscript{4}Assistant Professor, JCOE, Department Of Computer Engineering, Kuran

\textbf{Abstract}: For enabling communication and the personal information management from anywhere, an inextricable part of daily life is the Internet services and applications. To entertain this increase in application and complexity of data, web services have moved to a multi-tiered design wherein the application front-end logic runs by the web server and to a database or file server the data are outsourced. In this system, we propose an IDS system which is Double Guard, that models the network user sessions behaviour across both the back-end database as well as the front-end web server. By monitoring both web and subsequent database requests we are able to ferret out attacks that independent IDS would not be able to identify. Furthermore, in terms of sessions of training and functionality coverage we measure any multitier IDS limitations. With MySQL and lightweight virtualization with the help of an Apache web server we implement Double Guard. In both dynamic and static web applications the real-world traffic then processed and collected over a 15-day of system deployment period. Finally, using Double Guard, we will able to display a wide range of attacks which will give the 100 percent accuracy while for web services which are static, 0 percent false positives maintain by it and for web services which is dynamic, it is 0.6 percent false positives.

\textbf{1. Introduction}

Internet services and applications have become a part of daily life, enabling communication and the management of personal information from anywhere. Daily tasks, such as banking, are all done via the web. Such services typically employ a webserver front end that runs the application user interface logic, as well as a back-end server that consists of a database or file server. Due to their ubiquitous use for personal and/or corporate data, web services have always been the target of attacks. To overcome such issues and to detect intrusion in a web service or in a network different intrusion detection systems are used. Intrusion Detection System (IDS) is a device or software application that monitors network and/or system activities for malicious activities or policy violations and produces reports to a Management Station.

\textbf{Objectives}
1. To provide the security to computer systems.
2. To detect Intrusions and his activities have been noted.
3. To provides high security since the usage of session for each subsequent web request.
4. To detect and prevent the attacks on host machine.

The main contribution of our system DoubleGuard are as follows:
1. Composes both web IDS and database IDS to achieve more accurate detection
2. It also uses a reverse HTTP proxy to maintain a reduced level of service in the presence of false positives.
3. Instead of connecting to a database server, web applications will first connect to a database firewall. SQL queries are analyzed; if they’re deemed safe, they are then forwarded to the back-end database server.
4. Green SQL software work as a reverse proxy for DB connections
5. Virtualization is used to isolate objects and enhance security performance.
6. CLAMP is an architecture for preventing data leaks even in the presence of attacks. It isolates the code at the web server layer and data at the database layer.
7. Open VZ (Open Virtualization) is based on the Linux kernel and operating system. Open VZ allows a physical server to run multiple isolated operating system instances, known as containers or Virtual Environments (VEs).

\textbf{Limitation of IDS}
1. Detecting newly published attacks or variants of existing attacks.
2. Individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them.
3. However, it is found that these IDS cannot detect cases wherein normal traffic is used to attack the web server and the database server.
Existing System

In existing systems individually, the web IDS and the database IDS can detect abnormal network traffic sent to either of them. These IDSs cannot detect cases where normal traffic is used to attack the webserver and the database server.

For example, if an attacker with non-admin privileges can log in to a webserver using normal-user access credentials, he/she can find a way to issue a privileged database query by exploiting vulnerabilities in the web server.

2. Literature Review

A network intrusion detection system can be classified into two types: signature detection and anomaly detection. Anomaly detection first requires the IDS to define the characterized correct and acceptable static from dynamic behaviour of the system. It is used to detect the abnormal behaviour of the system. We first define the normal behaviour of the system and create profile of the user. In early IDS system that use the independent IDS used. Double Guard use dependent IDS used. Double Guard use the container ID using this ID each user session is assigned to each id. Our approach does not require input validation, source code validation and know the application logic. Double Guard uses the light weight virtualization to create and destroy the container by using the tool open VZ. We identify the causal relationship between web server request and database request. Our approach dynamically generate new containers and recycle the used ones. CLAMP (Confidentiality to LAMP) is an sensitive data leakage prevention technique even in the presence of attack. Clamp guarantees that user sensitive data can only be accessed by code running on the behalf of different user. Whereas DoubleGuard focuses on modeling the mapping patterns between the HTTP request and the database queries to detect the malicious user session. CLAMP requires modification to the existing application code, and the Query Restrictor works as a proxy to mediate all database access requests. Moreover, resource requirements and overhead differ in order of magnitude. DoubleGuard uses process isolation whereas CLAMP requires platform virtualization, and CLAMP provides more coarse-grained isolation than DoubleGuard. However, DoubleGuard would be ineffective at detecting attacks if it were to use the coarse-grained isolation as used in CLAMP. Building the mapping model in Double Guard would require a large number of isolated web stack instances so that mapping patterns would appear across different session instances. In addition, validating input is useful to detect or prevent SQL or XSS injection attacks. This is orthogonal to the Double Guard approach, which can utilize input validate on as an additional defense. However, we have found that Double-Guard 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 web server).

Problem Statement

Lots of existing intrusion Detection System (IDSs) examines the 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 generates models of network behavior for both web and database network interactions. In such multi-tiered architectures, the back-end database server is often 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 mean to exploit the back end. In order to protect multi-tiered web services, an e-client system call as Intrusion detection system is needed to detect known attacks by matching misused traffic patterns or signature.

3. Proposed Work

We present Double-Guard, an IDS system that models the network behavior of user sessions across both the front-end web server and the back-end database. By monitoring both web and subsequent database requests, we are able to ferret out attacks that independent IDS would not be able to identify. Furthermore, we quantify the limitations of any multitier IDS in terms of training sessions and functionality coverage. Using Double-Guard, we were able to expose a wide range of attacks.

The success of a project is greatly due to the personal expertise and responsibility of each member, but also to an adequate communication, collaboration and co-operation between the individual team members. Often, a good work team performance also depends directly on the personal characteristics of each team member, such as social skills and personality traits, making these human characteristics of vital importance in projects where the interaction and communication between the team members are fundamental for the achievement of the final objective. Additionally, the emotional state of a person plays a critical role in rational decision making, perception, human interaction, and human intelligence, affecting its own performance and the performance of the whole team during the project. Intrusion detection systems have been widely used to detect the attacks which are known by matching
misused traffic patterns or signatures to protect the multi-tier web services. The abnormal network traffic which are send by the attacker to attack the server can be detected by the web IDS and the database IDS and prohibit to enter within the server. But, if the attacker uses the normal traffic to attack the web servers and database server then such type of attack cannot be able to detect by a IDSs. So there is a need to develop a system which handle all types of attacks. So we propose a DoubleGuard Intrusion Detection System which is used to detect the attacks in multitier web services. In this system of DoubleGuard we are creating normality model of isolated user sessions which include both the web front-end as HTTP and back-end as File or SQL for network transaction.

DoubleGuard - An Intrusion Detection System which manages both front and back end of the multi-tier design & exposes a wide range of attacks with 100% accuracy. DoubleGuard is a system used to detect attacks in multi-tiered web services. This 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.

**Advantages of Proposed System**

1. The proposed IDS will help to detect the intrusion on both front end (Web Server) and back end (Database Server).
2. Double Guard able to identify a wide range of attacks with minimal false positives.

In this project, to detect attacks in web services which are multi-tiered, the present Double Guard system is used. In this system, isolated user sessions normality models can create that include both the web front end (HTTP) as well as back end (File or SQL) transactions of network. For achieving this a technique which is light weight virtualization technique is used for assigning the web session of every users to a particular container, an environment of isolated virtual computing.

**Attack Scenario**

**Privilege Escalation Attack:**

![Privilege Escalation Attack](image1.png)

It shows how a normal user may use admin queries to obtain privileged information. Now suppose that an attacker logs into the web server as a normal user, upgrades his/her privileges, and triggers admin queries so as to obtain an administrator’s data. This attack can never be detected by either the web server IDS or the database IDS.

**Hijack Future Session Attack:**

![Hijack Future Session Attack](image2.png)

It illustrates a scenario wherein a compromised web server can harm all the Hijack Future Sessions by not generating any DB queries for normal user requests. This class of attacks is mainly aimed at the web server side. An attacker usually takes over the web server and therefore hijacks all subsequent legitimate user sessions to launch attacks. For instance, by hijacking other user sessions, the attacker can eavesdrop, send spoofed replies, and/or drop user requests.
Injection Attack:

Attacks such as SQL injection do not require compromising the web server. Attackers can use existing vulnerabilities in 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.

Direct DB attack:

It illustrates the scenario wherein an attacker bypasses the web server to directly query the database. An attacker could also 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 could detect neither. Furthermore, if these DB queries were within the set of allowed queries, then the database IDS itself would not detect it either. However, this type of attack can be caught with Double Guard approach

Algorithm
Step 1. Identify the input type of HTTP request whether it is a query or a request.
Step 2. Store the input in hash table as per their type AQ for query and for request AR.
Step 3. The key for hash table entry will be set as the input itself.
Step 4. Forward AQ and AR to virtual server to validate
Step 5. If attack identified then virtual system automatically terminate the HTTP request.
Step 6. Else HTTP request is forwarded to the original server.
Step 7. Display information.
Step 8. Stop.

For modeling a static website, we have used Static Model Building Algorithm. This algorithm takes the input of training data set and builds the mapping model for static websites. For each unique HTTP request and database query, the algorithm assigns a hash table entry, the key of the entry is the request or query itself, and the value of the hash entry is AR for the request or AQ for the query, respectively. The algorithm generates the mapping model by considering all three mapping patterns i.e. Deterministic Mapping, Empty Query Set, No Matched Request pattern.

Model Building Algorithm
Require: Training Data set, Threshold t
Ensure: The Mapping Model for static website.
1: for each session separated traffic Ti do
2: Get different HTTP requests r and DB queries q in this session
3: for each different r do
4: if r is a request to static file then
5: Add r into set EQS
6: else
7: if r is not in set REQ then
8: Add r into REQ
9: Append 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 session ID i to the set AQq with q as the key
14: for each distinct HTTP request r in REQ do
15: for each distinct DB query q in SQL do
16: Compare the set ARr with the set AQq
17: if ARr = AQq and Cardinality (ARr)¿ t then
18: Found a Deterministic mapping from r to q
19: Add q into mapping model set MSr of r
20: Mark q in set SQL
21: else
22: Need more training sessions
23: return False
24: for each DB query q in SQL do
25: if q is not marked then
26: Add q into set NMR
27: for each HTTP request r in REQ do
28: if r has no deterministic mapping model then
29: Add r into set EQS
30: return true

4. Conclusion
In this system, we are developing the Double guard an Intrusion Detection System which manages both front and back end of the multi-tier design & exposes a wide range of attacks with 100% accuracy. DoubleGuard is a system used to detect attacks in multi-tiered web services. This 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. With a lightweight virtualization, from each web server session this project accomplish this by confine the flow of information. Furthermore, we quantified the accuracy of detection in our approach when we try to model web requests of static as well as dynamic with the back end file system and queries of database.

5. Acknowledgement

We express our sincere thanks to our project guide Prof. Arya C.S. who always being with presence & constant, constructive criticism to made this paper. We would also like to thank all the staff of COMPUTER DEPARTMENT for their valuable guidance, suggestion and support through the project work, who has given co-operation for the project with personal attention. Above all we express our deepest gratitude to all of them for their kind-hearted support which helped us a lot during project work. At the last we thankful to our friends, colleagues for the inspirational help provided to us through a project work.

6. References