SECURING CLOUD-HOSTED APPLICATIONS USING ACTIVE DEFENSE WITH RULE-BASED ADAPTATIONS

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
VAIBHAV V AKASHE
Dr. Prasad Calyam, Thesis Supervisor
DECEMBER 2021

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

SECURING CLOUD-HOSTED APPLICATIONS
USING ACTIVE DEFENSE WITH RULE-BASED ADAPTATIONS

presented by Vaibhav Akashe, a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Prasad Calyam

Dr. Giovanna Guidoboni

Dr. Yaw Adu-Gyamfi

ACKNOWLEDGMENTS

I would like to thank Dr. Prasad Calyam for his great support throughout my entire Master's degree. I appreciate the outstanding opportunity given to me by Dr. Calyam to work in Virtualization, Multimedia and Networking (VIMAN) Lab with a talented group of individuals on a state-of-art research area and lab facilities. He has imparted a lot of his enthusiasm and love for this field to me, and I would like to thank him for all the guidance and direction that has proven crucial to me as a student of computer science and as a researcher. He has been very patient, supportive and encouraging throughout my time here, and his keen insight and out-of-box pedagogy has taught me how to approach scientific research with the excitement that it deserves. I would like to express my gratitude towards Dr. Giovanna Guidoboni and Dr. Yaw Adu-Gyamfi for their interest to be a part of my thesis committee.

I would also like to thank all the members of VIMAN Lab, especially Samaikya Valluripally, Roland Oruche, Mauro Lemus Alarcon, Roshan Neupane, Songjie Wang, who all have guided and helped me selflessly. Their dedication to the lab and their research is commendable. I am grateful for the opportunity that this department and University have offered to me at so many different levels. Next, I would like to thank my friends for making this journey exciting and joyful for me. And finally I would like to thank my parents for supporting and showing their confidence in me.

Vaibhav V Akashe

Contents

ACKNOWLEDGMENTS . . . . **ii**

LIST OF TABLES . . . . **v**

LIST OF FIGURES . . . . **vi**

ABSTRACT . . . . **viii**

1 Introduction **1**

1.1 Active Defense Overview 1

1.2 Need for Active Defense in cloud-hosted video-based applications . . 2

1.3 Need for Active Defense in cloud-hosted data driven applications . . 3

1.4 Thesis Outline 4

2 Related Work **6**

2.1 Intrusion Detection Systems (IDS) 6

2.2 Active Defense Schemes 7

3 Applications Background **8**

3.1 Interactive video based learning environment case study : vSocial . 8

3.1.1 Factors Impacting VRLE Applications 9

3.1.2 Application Adaptation Frameworks 10

3.2 Data Driven application case study : OHDSI 10

3.2.1 Application Threat Model 12

3.2.2 Risk Assessment 13

4 Active Defense Frameworks and Solution Approaches **14**

4.1 Rule-based 3QS-adaptation Framework for vSocial 14

4.1.1 Anomaly Event Monitoring Tool 15

4.1.2 Adaptation Decision Making 15

4.1.3 Adaptation Control 17

4.1.4 Priority-based Queuing Model 18

4.2 OHDSI-Dolus System Design based on Defense by Pretense Methodology 20

4.2.1 Ensemble Learning 21

4.2.2 OHDSI-Dolus Pretense Initiation and Maintenance 23

5 Active Defense Evaluation Results 25

5.1 Performance Evaluation of Rule-based 3QS Adaptation Framework 26

5.1.1 Adaptation Decision Making and Control Unit 27

5.1.2 Quantification of Cybersickness for 3QS Anomalies 29

5.1.3 Risk and Cost Aware Trade-off Analysis 30

5.1.4 Recommendations Based on Key Findings 31

5.2 Performance Evaluation of defense by pretense based OHDSI-Dolus system 32

5.2.1 Risk Assessment Results 34

5.2.2 Detection Results 35

5.2.3 Defense by Pretense Qualitative Evaluation 38

6 Conclusion 40

List of Tables

Table Page

4.1 Performance metrics of our queuing model. 19

5.1 Potential adaptation choices for different 3QS anomaly events. . . . 28

5.2 Cost-aware application performance analysis of adaptations chosen for 3QS anomaly events. 31

5.3 Recommendations based on risk level (R_l), Cost level (C_l), and control on cybersickness (ΔCS). 32

5.4 Threat events related to OHDSI application with NIST-based guideline [43] used for risk calculation. 34

5.5 Comparison of OHDSI-Dolus performance with state-of-the-art active defense mechanisms that have the potential to be used for protection of cloud-based healthcare data processing pipelines. 38

List of Figures

Figure Page

3.1 Proposed framework for security and privacy analysis of social VRLE applications in order to adapt the system design 9

3.2 Overview of the data pipeline orchestration built on top of the OHDSI on AWS infrastructure. 11

4.1 Rule-based 3QS-adaption framework for a social VRLE system. 15

4.2 Modeling stages of our proposed rule-based 3QS-adaptation framework as a queue. 18

4.3 Illustration of proposed OHDSI-Dolus system where an attacker is tricked by redirection of malicious traffic to a quarantine VM for pretense, while the legitimate users can access the OHDSI hosted data sets. 21

4.4 Sequence diagram of OHDSI-Dolus defense system interactions for network traffic analysis and attack detection, along with attacker quarantine and active defense through initiation and maintenance of pretense. 24

5.1 Experimental testbed setup for 3QS-adaptation framework evaluation using anomaly event monitoring and rule-based decision making. 27

5.2 Avg. Latency measured (in ms) for QoA anomaly, QoS anomaly scenario in different adaptation scenarios. 29

5.3 Risk evaluation associated with the best (BA1, BA2), worst (WA1) and combination of adaptations in controlling cybersickness for the given QoA and QoS anomaly event. 31

5.4 AWS testbed used to evaluate OHDSI-Dolus for targeted attacks. 33

5.5 Heat map visualization of risk levels for different STRIDE categories A - F. . . 35

5.6 Dolus mirroring OHDSI server traffic at different stages for analysis and detection as viewed with AWS Cloud Watch. 36

5.7 Quality of Detection (QoD) results of DDoS attack detection based on accuracy and time taken by state-of-the-art detection mechanisms in comparison with our OHDSI-Dolus. 37

ABSTRACT

Security cloud-based applications is a dynamic problem since modern attacks are always evolving in their sophistication and disruption impact. Active defense is a state-of-the-art paradigm where proactive or reactive cybersecurity strategies are used to augment passive defense policies (e.g., firewalls). It involves using knowledge of the adversary to create of dynamic policy measures to secure resources and outsmart adversaries to make cyber-attacks difficult to execute. Using intelligent threat detection systems based on machine learning and active defense solutions implemented via cloud resource adaptations, we can slowdown attacks and derail attackers at an early stage so that they cannot proceed with their plots, while also increasing the probability that they will expose their presence or reveal their attack vectors.

In this MS Thesis, we demonstrate the concept and benefits of active defense in securing cloud-based applications through rule-based adaptations on distributed resources. Specifically, we propose two novel active defense strategies to mitigate impact of security anomaly events within: (a) social virtual reality learning environment (VRLE), and (b) healthcare data sharing environment (HDSE). Our first strategy involves a "rule-based 3QS-adaptation framework" that performs risk and cost aware trade-off analysis to control cybersickness due to performance/security anomaly events during a VRLE session. VRLEs provide immersive experience to users with increased accessibility to remote learning, thus a breach of security in critical VRLE application domains (e.g., healthcare, military training, manufacturing) can disrupt functionality and induce cybersickness. Our framework implementation in a real-world social VRLE viz., vSocial monitors performance/security anomaly events in network data. In the event of an anomaly, the framework features rule-based adaptations that are triggered by using various decision metrics. Based on our experimental results, we demonstrate the effectiveness of our rule-based 3QS-adaptation framework in reducing cybersickness levels, while maintaining application functionality. Our second strategy involves a “defense by pretense methodology” that uses real-time attack detection and creates cyber deception for HDSE applications. Healthcare data consumers (e.g., clinicians and researchers) require access to massive, protected datasets, thus loss of assurance/auditability of critical data such as Electronic Health Records (EHR) can severely impact loss of privacy of patient’s data and the reputation of the healthcare organizations. Our cyber deception utilizes elastic capacity provisioning via use of rule-based adaptation to provision Quarantine Virtual Machines (QVMs) that handle redirected attacker’s traffic and increase threat intelligence collection. We evaluate our defense by pretense design by creating an experimental Amazon Web Services (AWS) testbed hosting a real-world OHDSI setup for protected health data analytics/sharing with electronic health record data (SynPUF) and publications data (CORD-19) related to COVID-19. Our experiment results show how we can successfully detect targeted attacks such as e.g., DDoS and create redirection of attack sources to QVMs.

Chapter 1

Introduction

1.1 Active Defense Overview

Active defense [1] is a proactive cybersecurity strategy that involves creation of dynamic management or even offensive measures to outsmart adversaries in order to make cyber-attacks difficult to execute. Using intelligent detection systems and defense solutions such as honeypots [2] and machine learning algorithms, active defense can be performed to slow down attacks and derail attackers at an early stage so that they cannot proceed with their plan, increasing the probability that they will expose their presence or reveal their attack vector. Thus, active defense schemes gain threat intelligence on targeted attacks and enable organizations to understand the nature of attacks, create robust defenses and also prevent recurrence of attacks.

1.2 Need for Active Defense in cloud-hosted video-based applications

Active defense mechanisms in cloud platforms need to be robust against targeted attacks (such as DDoS, malware and SQL injection) whose impact can be amplified due to the elastic resource nature of cloud platforms. Particularly, there are critical challenges in securing interactive video based learning environments. Studies such as [3] outlines interactive video based learning environments i.e., E-learning, a novel way in the learning process that involves more interaction between the learners and teachers to some extent network environment and is a honey pot to attract many attackers and it may have some potential security risks such as: malicious attacks, hackers and so on. Another such example is Virtual Reality based Learning Environments (VRLEs) such as vSocial [4] for youth with learning disabilities. With the dynamic user-system interactions for content rendering, VRLEs are a target for an attacker to trigger security attacks [5], [7]. In addition, the work in [8] details about the performance issues that can disrupt the social VRLE user experience. However, prior works lack in the knowledge to address both performance and security issues that can impact the user experience and user safety in VRLE sessions. Failure to address such impediments can lead to deface attacks on the VR content with offensive images [9] that can hamper user experience. They can also lead to application latency issues that degrade performance. Based on prior works in VRLE and other IoT applications [10], [11] we adopt the following definitions of various performance ($3Q$) factors: Quality of Application (QoA) – a measure of the application performance; Quality of Service (QoS) – a measure of network resources such as bandwidth and jitter; Quality of Experience (QoE) – a measure of the perceived satisfaction or annoyance of a user's experience. Similarly, we adopt the definition of security – as a condition that ensures a VR system is able to perform critical application functions with the establishment of confidentiality, integrity, and availability [7]. Together, such performance and security issues can induce "cybersickness" (e.g., eyestrain, nausea, headache, disorientation of user movement) [12, 13]. Hence, there is a need to study methods to mitigate impact of performance and security anomaly events that induce cybersickness.

1.3 Need for Active Defense in cloud-hosted data driven applications

As outlined above, there are critical challenges in securing data driven applications such as healthcare applications to avoid issues with availability or data breaches/loss, while also providing solutions that are cost effective, efficient, and timely [14]. Healthcare applications with data processing pipelines handle critical data such as Electronic Health Records (EHR), and sensitive personal health related data generated through medical devices. There have been prior works on securing EHR data in cloud-based platforms using Blockchain-based solutions [15] or through access control mechanisms based on the lattice model [16]. There have been proposals for attribute-based encryption access control, homomorphic encryption, and storage path encryption to improve privacy and security in healthcare applications [17]. However, to the best of our knowledge, none of the prior works have focused on active defense involving making use of dynamic management or offensive strategies, particularly relating to healthcare data processing pipelines orchestrated in cloud platforms.

In this work, we demonstrate the concept and benefits of active defense in securing cloud-based applications through rule-based adaptations on distributed resources. Specifically, we propose two novel active defense strategies to mitigate impact of security anomaly events within: (a) social virtual reality learning environment (VRLE), and (b) healthcare data sharing environment (HDSE). Our first strategy involves a "rule-based 3QS-adaptation framework" that performs risk and cost aware trade-off analysis to control cybersickness due to performance/security anomaly events during a VRLE session. VRLEs provide immersive experience to users with increased accessibility to remote learning, thus a breach of security in critical VRLE application domains (e.g., healthcare, military training, manufacturing) can disrupt functionality and induce cybersickness. Our framework implementation in a real-world social VRLE viz., vSocial monitors performance/security anomaly events in network data. In the event of an anomaly, the framework features rule-based adaptations that are triggered by using various decision metrics. Based on our experimental results, we demonstrate the effectiveness of our rule-based 3QS-adaptation framework in reducing cybersickness levels, while maintaining application functionality. Our second strategy involves a "defense by pretense methodology" that uses real-time attack detection and creates cyber deception for HDSE applications. Healthcare data consumers (e.g., clinicians and researchers) require access to massive, protected datasets, thus loss of assurance/auditability of critical data such as Electronic Health Records (EHR) can severely impact loss of privacy of patient's data and the reputation of the healthcare organizations. Our cyber deception utilizes elastic capacity provisioning via use of rule-based adaptation to provision Quarantine Virtual Machines (QVMs) that handle redirected attacker's traffic and increase threat intelligence collection. We evaluate our defense by pretense design by creating an experimental Amazon Web Services (AWS) testbed hosting a real-world OHDSI setup for protected health data analytics/sharing with electronic health record data (SynPUF) and publications data (CORD-19) related to COVID-19. Our experiment results show how we can successfully detect targeted attacks such as e.g., DDoS and create redirection of attack sources to QVMs.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: In Chapter 2, we describe the thesis related work on Intrusion Detection Sysytems (IDE) and active defense. In Chapter 3, we describe background that provide context to the solution approach.

Chapter 4 we elaborate on our solutions and provide a detailed description of our approaches with reference architectures. Chapter 5 evaluates the effectiveness of our frameworks. Finally, Chapter 6 concludes the thesis.

Chapter 2

Related Work

2.1 Intrusion Detection Systems (IDS)

Many prior works have addressed detecting security threats in cloud environments by using a variety of IDS techniques that utilize pattern recognition and machine learning concepts. The study presented in [18] provides an extensive review on cloud computing focusing on security gaps, and proposes a proactive machine learning based threat detection model. Similarly, authors in [20] propose a learning-based IDS to detect network-based intrusion in cloud platforms. Particularly, DDoS attacks pose serious threats to cloud-hosted services. Studies presented in [21, 22, 23] detail detection of DDoS attacks, and studies in [24, 25, 26, 27] present techniques for detection of APTs, including attacks such as malware, botnets, data breach and data scraping.

Coupled with these emerging techniques, IDSes tend to be effective against detecting targeted attack threats. The work in [19] presents a comprehensive review about IDS and a study in [28] outlines current need for an advanced novel IDS approach additionally studies in [29] detail many host-based and network-based IDS techniques that are widely used by enterprises in both their data centers, as well as in their cloud-hosted application environments.

2.2 Active Defense Schemes

There are many prior works involving different kinds of cyber deception to trick the attacker and defend against threats. For example, the work presented in [29] proposes the use of system agents to launch on-demand honeypot VMs with enhanced VM introspection techniques. Another study in [32] uses the concept of a 'honey patch' to make a patched server reply to an adversary in a similar fashion to the way a non-patched server would. It then produces a container that appears to be a vulnerable system – but with redacted information hidden from the adversary, which helps to avoid leaking of sensitive information.

Studies such as those in [30] propose prevention strategies against DDoS attacks targeting eHealth clouds. Their approach involves detection of malicious activity to alert system administrator and subsequently blacklist the attacker's source address to block communications from the adversary. Similarly, the study in [31] proposed an active defense mechanism against data ex-filtration attacks in SaaS clouds by using a technique that matches the default identifier i.e., MAC address with the embedded identifier within the file. If the MAC address does not match, a corresponding decoy document (i.e., a honey file) is returned. Additionally, the framework in [33] involves an active defense strategy that uses decoys of real system components to obfuscate the network and in turn make it harder for a potential adversary to identify the real components.

Chapter 3

Applications Background

3.1 Interactive video based learning environment case study : vSocial

Social Virtual Reality Learning Environments (VRLEs) are a convergence of virtual reality (VR), Internet-of-Things (IoT) and cloud computing technologies [38]. As shown in Figure 3.1, they integrate real-world smart things (i.e., VR headsets/glasses) with virtual objects/avatars for a real-time immersive interaction of geographically distributed users [53]. Social VR applications in education or collaborative tasks adopt virtual worlds as learning environments [54], where participants can interact effectively with higher engagement and performance [55]. To facilitate continuous interaction between the users (e.g., instructors and students), the networked VRLE components collect data from distributed user locations, and seamlessly integrate web-based tools to render VRLE content. However, such capabilities in these socio-technical systems demand for high-performance and robust VRLE application features.

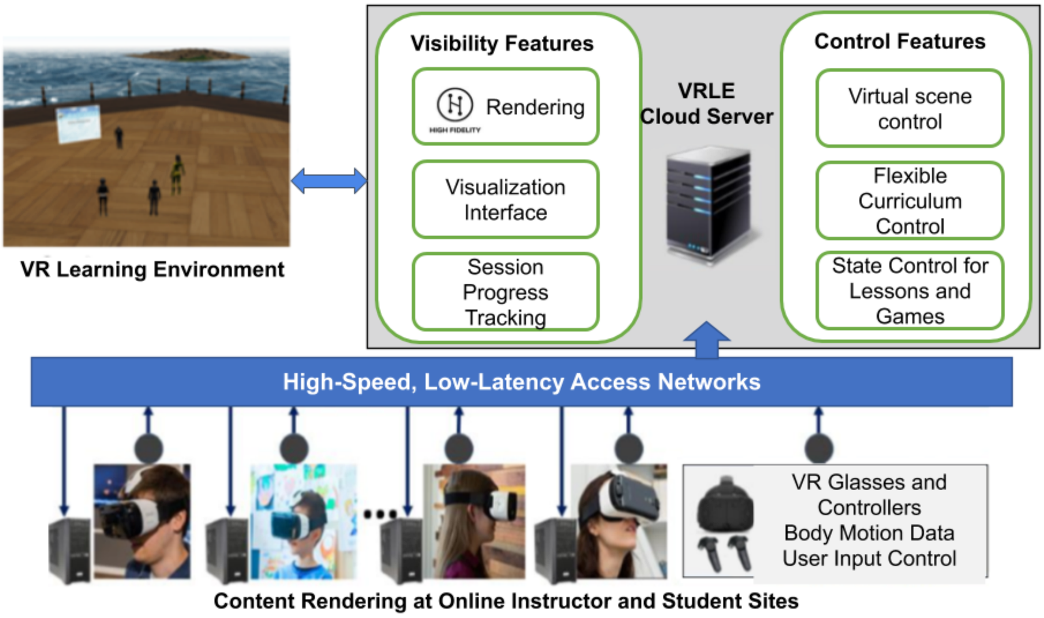
VR Learning Environment
Visibility Features
Rendering
HIGH FIDELITY
Visualization Interface
Session Progress Tracking
VRLE Cloud Server
Control Features
Virtual scene control
Flexible Curriculum Control
State Control for Lessons and Games
High-Speed, Low-Latency Access Networks
VR Glasses and Controllers
Body Motion Data
User Input Control
Content Rendering at Online Instructor and Student Sites

Figure 3.1: Proposed framework for security and privacy analysis of social VRLE applications in order to adapt the system design

3.1.1 Factors Impacting VRLE Applications

Prior works [5, 7, 8] addressed performance and security issues in social VRLE applications. The work in [5] described potential security, privacy and safety issues that can trigger disruption in the VRLE application functionality. In addition, the work in [7] also detailed vulnerable components in VRLE that can lead to sophisticated cyber-attacks such as Loss of Integrity and privacy leakage. Authors in [8] model performance issues via a 3Q-model to determine the causes of disruption of VRLE user experience.

The impact of such effects can specifically induce cybersickness, thus compromising *user safety* in a VRLE session [45, 46]. On the other hand, works related to other applications such as remote instrumentation [56] and video-based cloud applications [57] analyze performance factors that disrupt user experience and propose a 3Q factors interplay model for determining suitable adaptations. Using the outlined security and performance issues of VRLE in the above state-of-the-art, we propose a continuous 3QS anomaly event monitoring approach to guide adaptation control decisions to minimize cybersickness levels during a VRLE session.

3.1.2 Application Adaptation Frameworks

There have been works [58, 59] that address either performance or security issues in the context of a control-feedback scheme to adapt cloud-based IoT applications. For instance, the works in [58, 59, 60] present solutions that feature adaptive control mechanisms to address scalability and latency issues based on user's service level objective (SLO) and cost constraints. Adaptive control mechanisms [61] related to addressing security issues at the application layer have been studied at an on-demand resource management level involving e.g., DoS attacks [62]. In contrast, our 3QS-adaptation framework considers the interplay of security and performance factors potentially inducing cybersickness. Our adaptations consider time-sensitive response of the system by using performance metrics such as: response time, resource usage for an adaptation and risk of performing that adaptation along with the cost constraint for a given performance/security issue.

3.2 Data Driven application case study : OHDSI

Healthcare data consumers (e.g., clinicians and researchers) require access to massive datasets which are usually residing in multiple and disparate data sources. This creates many challenges for the data consumers to access and compile the data required to conduct research and make timely decisions. Data processing pipelines are increasingly being used to combine data from multiple sources, allow access to multiple users, and include multiple data analytic tools to orchestrate data aggregation, processing and visualization processes. To facilitate the orchestration of such data pipelines, exemplar technologies, such as OHDSI [51] have been adopted for use in cloud environments by healthcare organizations.

To develop our network-based active defense solution, we deployed the open-source OHDSI on the AWS platform as illustrated in Figure 3.2. The OHDSI on AWS deployment provides an enterprise class, multi-user, and scalable healthcare data sharing and analytics functionality [37]. Its main components include a Com-

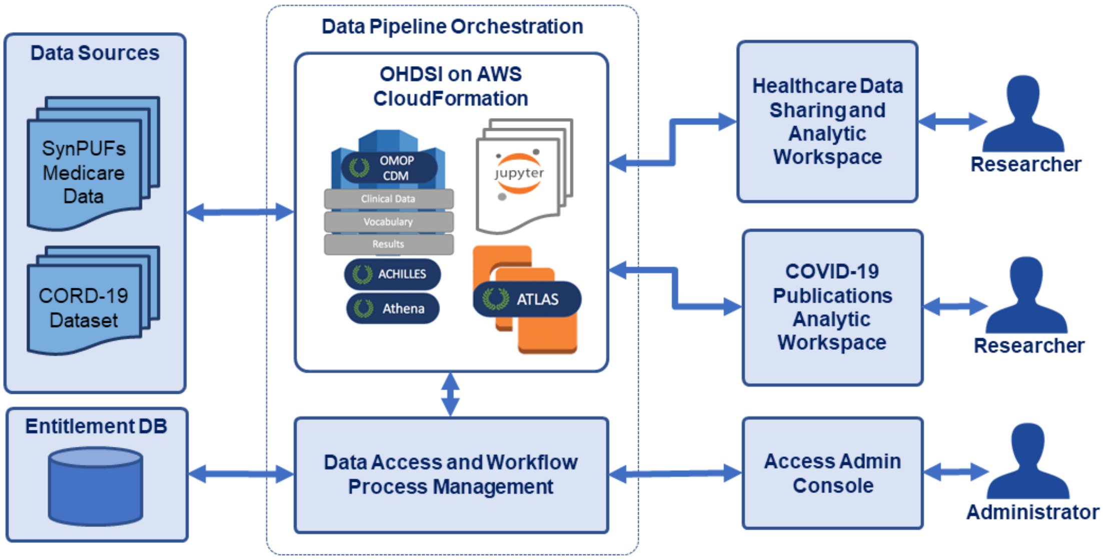
Data Sources
SynPUFs Medicare Data
CORD-19 Dataset
Entitlement DB
Data Pipeline Orchestration
OHDSI on AWS CloudFormation
OMOP CDM
Clinical Data
Vocabulary
Results
ACHILLES
Athena
jupyter
ATLAS
Data Access and Workflow Process Management
Healthcare Data Sharing and Analytic Workspace
COVID-19 Publications Analytic Workspace
Access Admin Console
Researcher
Researcher
Administrator

Figure 3.2: Overview of the data pipeline orchestration built on top of the OHDSI on AWS infrastructure.

mon Data Model (CDM) based on the OMOP-CDM schema, which is deployed on an AWS Redshift data warehouse. The CDM schema allows the integration of disparate data-sources into a common format (model) and common representation (terminology, vocabulary, coding), allowing the definition and execution of standard analytic processes. Other OHDSI components include out-of-the-box open-source analytic tools such as: (i) ATLAS, a web-based application for researchers to conduct analyses on data loaded to the OMOP-CDM through creation of cohorts based on drug exposure or diagnosis of a particular condition. The cohort results are visualized in the tool's user interface, or stored in a relational repository to be used by other analytic tools; (ii) ACHILLES, an application used to analyze the database hosting the CDM and evaluate data quality; (iii) ATHENA, a tool that is used to generate and load standardized data vocabularies into the CDM repository. Once data is available in the CDM, evidence knowledge can be generated using the included analytic tools and models available in the workspace available via Jupyter Notebooks or R-Studio.

3.2.1 Application Threat Model

To better understand the threats and their imposed risks to the OHDSI application use cases, we use the Microsoft STRIDE methodology [52] to create an application-level threat risk model. We organised the attacks against the OHDSI application into Loss of users' trust, Loss of confidentiality, Loss of availability and Loss of integrity.

- **S**poofing: IP spoofing Example - An attacker can alter the IP packet to gain access to the OHDSI application server as an authorized user. Successful IP spoofing attack can cause loss of trust for users and loss of confidentiality in the OHDSI system.
- **T**ampering: Data Alteration Example - Malicious user can spoof queries to retrieve and modify data and can cause loss of integrity in the OHDSI system.
- **R**epudiation: Example - Attacker can impersonate a user to retrieve and modify data that can lead to loss of confidentiality and integrity for OHDSI system users.
- **I**nformation Disclosure: SQL injection/Malware infection Example - Attacker can perform a SQL injection attack to affect the database or gain access to unauthorized data. Also presence of malware on system can lead to leakage of users' data. Such attacks can cause loss of confidentiality and integrity for OHDSI application users.
- **D**enial of Service: Example - Attacker can perform multiple SQL queries to overwhelm the database system, which can lead to loss of availability for OHDSI users.
- **E**levation of Privilege: Data Tampering Example - An attacker can tamper data or even delete data on the network, which can then lead to loss of integrity for data, and loss of availability for the OHDSI users.

Exploitation of potential vulnerabilities such as DDoS, Malware/SQL injection identified by our threat model pose major threats to healthcare data processing pipelines. These vulnerabilities may result in possible risks to patient safety and theft or loss of health related information, which have serious consequences in the healthcare organization operations.

3.2.2 Risk Assessment

Following the threat modeling study performed using the STRIDE methodology, we use the methodology in the NIST risk assessment guideline [43] to calculate the potential risk levels for various threats impacting the OHDSI in AWS application. The NIST methodology populates the impact values and likelihood values for specific threats being considered. The impact values are derived from assessed potential impact resulting from a compromise of the confidentiality, integrity, or availability for any information type due to security threats. The likelihood values are a weighted factor based on a subjective analysis of the probability that a given threat is capable of exploiting a given vulnerability of the threats. Following this, the overall risk values are calculated by factoring the likelihood and impact scores, which are finally normalized into a quantitative scale of 0-10. These ranges for scales are: 9-10 indicating very high risk, 7-8 indicating high risk, 4-6 indicating moderate risk, 1-3 indicating low risk, and 0 indicating very low level of risk. We evaluate the risk levels for different threat events in the STRIDE model based on the NIST methodology and present the details of the results in Section V (Performance Evaluation) of this paper. Our risk assessment guides the design principles for our Dolus-OHDSI active defense system design.

Chapter 4

Active Defense Frameworks and Solution Approaches

In this chapter, we present the overview of our two novel active defense strategies to mitigate impact of security anomaly events within: (a) social virtual reality learning environment (VRLE), and (b) healthcare data sharing environment (HDSE). Our first strategy involves a "rule-based 3QS-adaptation framework" that performs risk and cost aware trade-off analysis to control cybersickness due to performance/security anomaly events during a VRLE session. Our second strategy involves a “defense by pretense methodology” that uses real-time attack detection and creates cyber deception for HDSE applications.

4.1 Rule-based 3QS-adaptation Framework for vSocial

In this section, we present the overview of our novel rule-based 3QS-adaptation framework as described in [63], to control the impact of cybersickness levels in VRLE as shown in Figure 4.1.

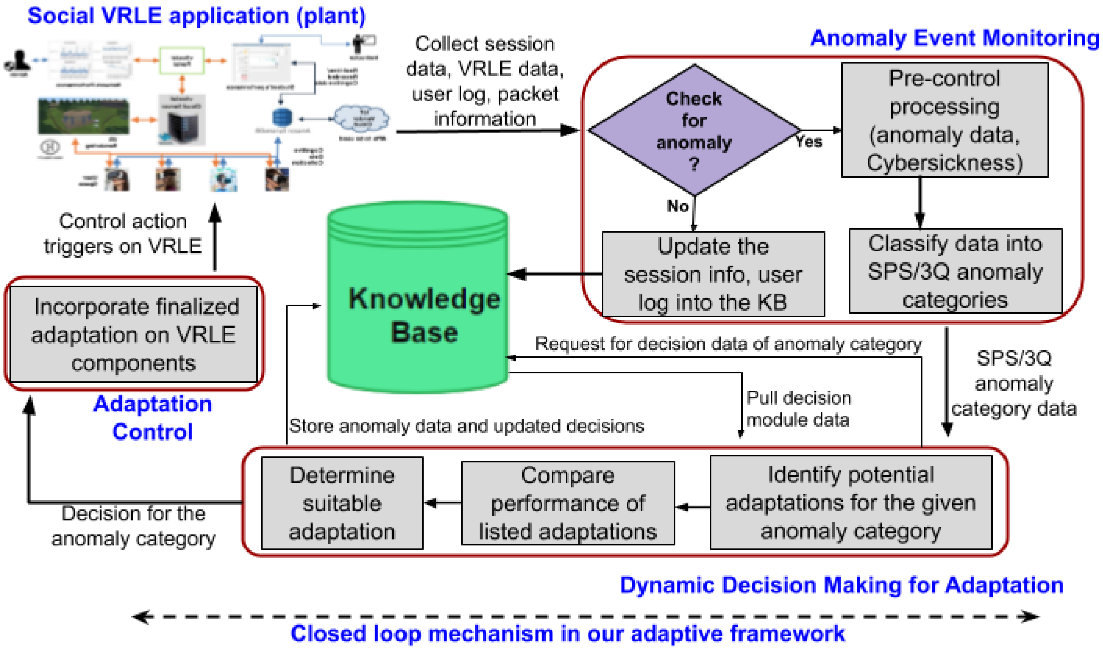
Social VRLE application (plant)
Collect session data, VRLE data, user log, packet information
Anomaly Event Monitoring
Check for anomaly ?
Yes
No
Pre-control processing (anomaly data, Cybersickness)
Update the session info, user log into the KB
Classify data into SPS/3Q anomaly categories
Knowledge Base
Control action triggers on VRLE
Incorporate finalized adaptation on VRLE components
Adaptation Control
Request for decision data of anomaly category
SPS/3Q anomaly category data
Pull decision module data
Store anomaly data and updated decisions
Determine suitable adaptation
Compare performance of listed adaptations
Identify potential adaptations for the given anomaly category
Decision for the anomaly category
Dynamic Decision Making for Adaptation
Closed loop mechanism in our adaptive framework

Figure 4.1: Rule-based 3QS-adaption framework for a social VRLE system.

4.1.1 Anomaly Event Monitoring Tool

To identify any potential 3QS issues in a social VRLE, we developed an anomaly event monitoring tool [34] to observe the network behavior changes, and user activity trends during the VRLE session. We create alarms to trigger when an anomalous behavior pattern is identified in the vSocial application. The anomaly event types include: QoA issues (e.g., visualization delay due to network lag), QoS issues (e.g., packet loss), and security issues (e.g., DoS attack, unauthorized access). Next, we collect this anomaly event data as shown in Figure 4.1 in order to calculate the corresponding impact on the cybersickness level for the session user(s). Following this, we classify the collected anomaly event data into specific 3QS categories.

4.1.2 Adaptation Decision Making

The anomaly event data is classified based on 3QS issue categories. Given anomaly event, our 3QS-adaptation framework activates a decision module that has knowledge of potential adaptations for the specific event as shown in Figure 4.1. Each of

Algorithm 1: Build Decision Units

```
Input: Anomaly type list
Output: Relevant Decision units
begin
    Function BuildAdaptation ():
        for each $AnomalyType \in AnomalyTypeList$ do
            Function BuildDecisionUnits ():
                Let $TupleList = [\,]$
                for each $Adaptation \in TupleList$ do
                    Let $Tuple = (A_n, Ct, I)$ $TupleList.append(Tuple)$
                end
                return $DecisionUnit\{AnomalyType_i, TupleList\}$
            end Function
        end
        return $Adaptation\{DecisionUnit_0, ..., DecisionUnits_k\}$
    end Function
end
```

the detected anomaly event categories are sent as input to the Algorithm 1 which details the functionality of the decision module. The decision module allows it to compare an anomaly event in a particular category with a set of relevant decision units as described in Algorithm1. Each decision unit has the knowledge on how to deal with a specific type of anomaly event i.e., decision units contain a list of potential candidate adaptations that are retrieved from the knowledge base module. The function *BuildDecisionUnits()* in Algorithm 1 describes how decision units are developed where, each decision unit contains a list of defined tuples. These tuples are of the form $\{A_n, Ct, I\}$, where A_n represents the adaptation name, Ct represents the history of adaptation in terms of number of times that specific choice was implemented, and I represents the impact on cybersickness level after the adaptation was implemented for a given anomaly event. As and when such decision units are created, our decision module retrieves the decision units using the *BuildAdaptation()* function in Algorithm 1.

Next, the decision module traverses through the list of candidate adaptations in each of these retrieved decision units to determine the most suitable adaptation. Each of the listed decision units will be sorted using the order of attributes I, Ct in tuples which are termed as “decision metrics” along with the reduced response time taken by a specific adaptation. The head of the sorted list of candidate adaptations represent the most suitable adaptation for a given anomaly event. With every iteration of handling anomaly events, the Ct value related to the considered adaptation gets updated into the knowledge base. Thus, using the decision module, our 3QS-adaptation framework facilitates dynamic decision making for a suitable adaptation to reduce the induced cybersickness level for a given anomaly event.

Our proposed 3QS-adaptation framework stores the baseline data of benign application behavior into the knowledge base for handling future anomaly events in a social VRLE. The knowledge base actively stores VRLE session information, detected anomaly event patterns along with the potential adaptations and associated user data. The anomaly event traces in the knowledge base can be helpful to a network/system administrator to determine the causes of the detected anomalies, and improve the effectiveness of the adaptations. Moreover, the knowledge base can be used as a medium for threat intelligence collection to train our decision module for mitigation of zero-day 3QS anomaly events that can arise in an individual scenario and/or in combination scenarios.

4.1.3 Adaptation Control

The control module in our 3QS-adaptation framework enacts suitable adaptations for a given anomaly event category. Once the decision outcome (i.e., suitable adaptation) is obtained, the control module first calculates the risk level associated to a choice of adaptation, along with the cost incurred to control the induced cybersickness anomaly event. Next, the control module invokes an action using an alarm (using e.g., AWS CloudWatch) for the relevant functionality of the determined adaptation. In addition, the risk and cost aware decision outcome implementation is evaluated for the feedback (e.g., control on cybersickness level, user satisfaction). If the anomaly event is successfully handled, then this session information along with the control module data is updated into the knowledge base for handling similar future anomaly events. Thus, the anomalies are monitored continuously, and we perform dynamic decision making to invoke the suitable control actions iteratively for on-demand resource provisioning that delivers satisfactory user experience and controls cybersickness levels in a social VRLE.

4.1.4 Priority-based Queuing Model

In our 3QS-adaptation framework, the entire timeline of anomaly event processing can be divided into three parts each considered at *VRLE application plant*, *anomaly monitoring tool* and *decision module* as shown in Figure 4.1. This behavior of anomaly event data processing represents a queue, and thus we model our framework into a *M/M/1/K* finite queuing system to capture the pattern of VRLE application performance. This analytical model is based on an embedded Markov Chain, featured by states, events, transitions. The requests that enter into the queue are the anomaly events caused by 3QS issues, which are processed mainly on a priority basis i.e., in the order of events that have the ability to cause higher cybersickness levels. We focus especially on the response time in addressing the anomaly events inducing cybersickness.

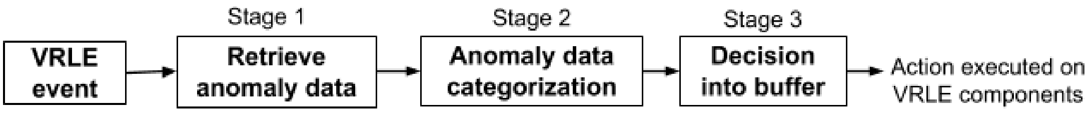

Figure 4.2: Modeling stages of our proposed rule-based 3QS-adaptation framework as a queue.

The processing of an incoming request includes three stages: *stage 1* (collecting anomaly event data), *stage 2* (categorization into anomalies caused by 3QS issues), and *stage 3* (anomaly event data pushed into the decision module) as shown in Figure 4.2. After *stage 3*, the processed event record leaves the queue, where the anomaly data is sent to the decision module to determine the suitable action on the corresponding VRLE component. Each stage described in Figure 4.2, has a different average service rate, represented as μ_1, μ_2, and μ_3. Thus, the overall

Table 4.1: Performance metrics of our queuing model.

No. of events in queue	W_q **(in sec)**	$\bar{X}$ **(in sec)**	R_s **(in sec)**	**No. of processed severe anomalies**
10	2400.48	0.146	3300	4
20	5700	0.204	6303.98	5
30	8700	0.28	9304.15	7
40	11700	0.36	12304.32	11

response time of the system in processing one data record can be computed by solving the markov chain transition model. In this process, the execution of the three stages is mutually exclusive, which means that the second record will not be processed until the previous one is completed. We assume the processing times at each stage is exponentially distributed, and the data retrieval at stage 1 follow a Poisson arrival with an expected rate of λ.
The mean response time (RT_q) to process an anomaly event in the queue can be obtained by using the Little's formula [8].

The wait time of the queue W_q is derived based on the number of events in the queue L_q and arrival rate (λ)

$$W_q = \frac{L_q}{\lambda} \tag{2}$$

$\bar{X}$ is the sum of the mean service time for all three stages, and can be written as -

$$\bar{X} = \sum_{n=1}^{3} 1/\mu_n \tag{4}$$

We use the above analytical model in the performance evaluation experiments to determine the waiting delays that might occur in processing the anomaly events inducing cybersickness. To elucidate, a low cybersickness inducing anomaly trigger can be delayed, while a severe threat posing anomaly trigger can be urgently

handled by allowing it to experience lower wait times in the triggers handling queue. To achieve such a handling, we use our priority queue model as a Binary-Heap [50] to perform reheapficiation of the events in the queue once an anomaly event is deleted from the queue.
Using the above formulation, Table 4.1 lists the calculation of the overall system response time (R_s) as Sum (RT_q, R_{at}), where RT_q is the response time in queue and R_{at} is the time taken for an adaptation to implement. In addition, we also enlist the number of processed severe anomaly events (i.e., with high cybersickness level) for a given number of anomaly events in the queue in Table 4.1.

4.2 OHDSI-Dolus System Design based on Defense by Pretense Methodology

Our work builds upon prior work on the Dolus 'defense by pretense' system [35] which sits on a cloud network to perform tasks to intelligently detect and mitigate targeted attacks such as DDoS and Advanced Persistent Threats (APTs) in cloud platforms. Dolus uses a 'defense by pretense' active defense strategy that creates cyber deception by leading attackers into experiencing a false sense of success while a robust co-operative defense solution is being designed to mitigate attack impact or even dis-incentivize the attacker to continue a targeted attack. The cyber deception utilizes elastic capacity provisioning via use of Quarantine Virtual Machines (QVMs) that handle redirected attacker's traffic and increase threat intelligence collection.

Figure 4.3 shows the OHDSI-Dolus system as described in [64], with physical architecture components for initiation and maintenance of pretense in the event of a targeted attack. The OHDSI-Dolus takes advantage of elastic compute services provided by cloud service providers, particularly the application load balancer and on-demand provisioning of virtual instances. We place the application load

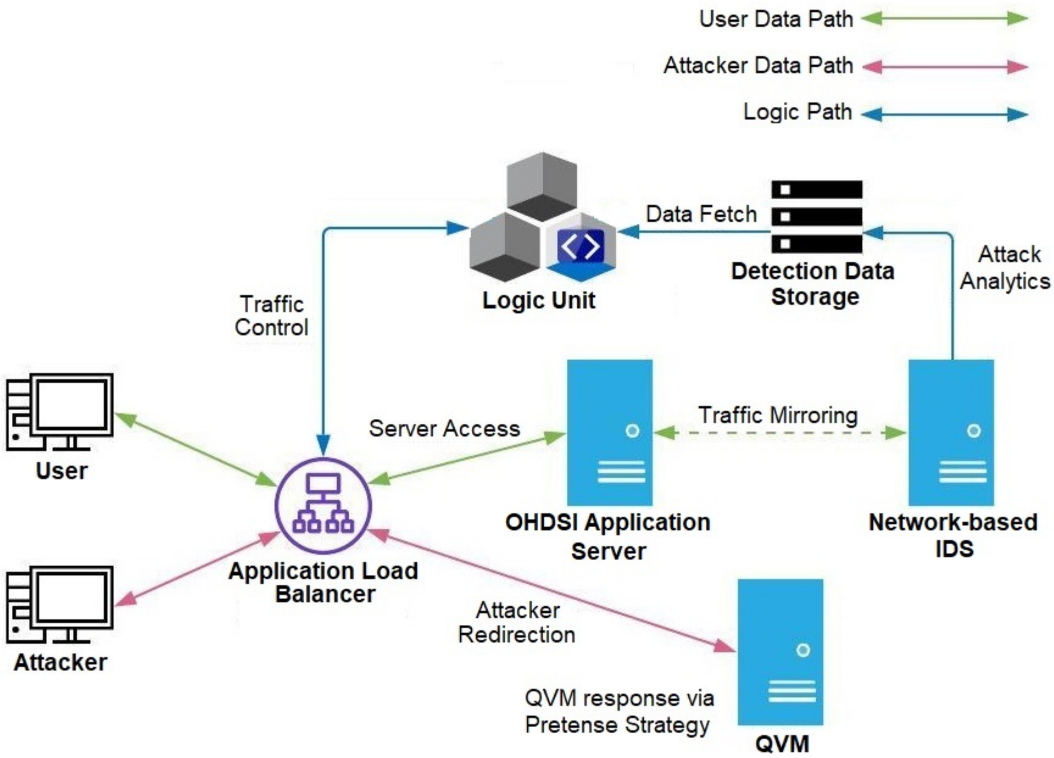

Figure 4.3: Illustration of proposed OHDSI-Dolus system where an attacker is tricked by redirection of malicious traffic to a quarantine VM for pretense, while the legitimate users can access the OHDSI hosted data sets.

balancer in between user and the cloud resources that are deployed in a virtual private cloud (VPC).

When the IDS suspects a network intrusion or cyber-attack event, then a *Quality of Detection* (QoD) protocol gets triggered in the OHDSI-Dolus system. If QoD value is above a certain threshold which validates that an intrusion or cyber-attack event has indeed been detected accurately, then the Dolus pretense is initiated and maintained.

4.2.1 Ensemble Learning

We use the ensemble learning methodology to determine the accuracy of our attack detection. Network traffic is collected through the network-based IDS in OHDSI-Dolus when legitimate users and attackers try to access the cloud-hosted OHDSI services. Users interact with the OHDSI application server by requesting different healthcare related data resources. We monitor the attack traffic targeted to the data processing pipeline server and capture e.g., bytes transmitted, number of packets, source, and destination IP address. Subsequently, the QoD is calculated by taking attack-related factors into consideration as well as the based on the complexity/effectiveness of the detection mechanisms evidenced by e.g., data sanity, and detection time/accuracy. The formula to calculate the QoD value is as follows:

$$QoD = \frac{1}{n}\sum_{i=1}^{n}\frac{a_i}{t_d} \tag{4.1}$$

In the QoD formula, the ($a_i \in [0,1]$) refers to the accuracy of the ensemble learning model (dependent also on data sanity) used to identify the cyber-attacks. t_d is the time taken (in seconds) for the machine learning model to detect the attacks and n represents number of test iterations in the evaluation. These QoD values ranges from 0 to 100, hence we normalize these values into [0,10] range by dividing the values by 10. If QoD values are above non-zero, the pretense initiation and maintenance is invoked, however the administrator may set a higher threshold as suited in accordance with the active defense policies of the healthcare organization.

We use Frenetic (an open-source software-defined network controller platform [35]) to execute Python scripts that identify suspicious packets, gather attack patterns in order to redirect packets to pertinent QVMs. IP addresses of the attackers are then blacklisted by updating a corresponding network policy. We characterize the attack data for DDoS by measuring e.g., the total bytes transferred, rate of transfer, connections made, and attack duration. This allows us to get dynamic “suspiciousness scores” of attackers and their domain nodes for targeted attacks. To emulate a DDoS attack, we exhaust the targeted application using a SlowHTTPTest [36] and thereby cause random changes in e.g., number of packets, and attack times. We also perform event-based simulations to get different suspiciousness scores for attacks as follows:

Destination suspiciousness for trace t:

$$dst_i = w_{dst} \times \frac{numDst_i - numDstMin_i}{numDstMax_i - numDstMin_i} \tag{4.2}$$

Flow suspiciousness for trace t:

$$flows_i = w_{flows} \times \frac{numFlows_i - numFlowsMin_i}{numFlowsMax_i - numFlowsMin_i} \tag{4.3}$$

Bytes suspiciousness for trace t:

$$\begin{gathered} bytes_i = w_{bytes} \times \frac{numBytes_i - numBytesMin_i}{numBytesMax_i - numBytesMin_i}; \\ w_{dst} \in [0.0, 1.0]; w_{flows} \in [0.0, 1.0]; w_{bytes} \in [0.0, 1.0] \end{gathered} \tag{4.4}$$

Device suspiciousness for trace t:

$$ss_i = \sqrt{\frac{dst_i^2 + flows_i^2 + bytes_i^2}{3}} \tag{4.5}$$

We calculate the *ss* values based on the captured network traces using three main features: destinations, flows, and bytes. For each attacker node i on the network, and for trace t, we assume the weight parameters i.e., w_{dst}, w_{flows}, w_{bytes} to be equal to 1 in a general case of suspiciousness score calculations. Also, the *Min* and *Max* values are assumptions made per attack nodes based on the expected behavior of the network flows corresponding to user hosts' traffic.

4.2.2 OHDSI-Dolus Pretense Initiation and Maintenance

The entire procedure of our OHDSI-Dolus interactions involving sequential steps are shown in Figure 4.4 for classification of user traffic and attacker traffic as well as creation of the attacker quarantine with active defense through initiation and maintenance of pretense.

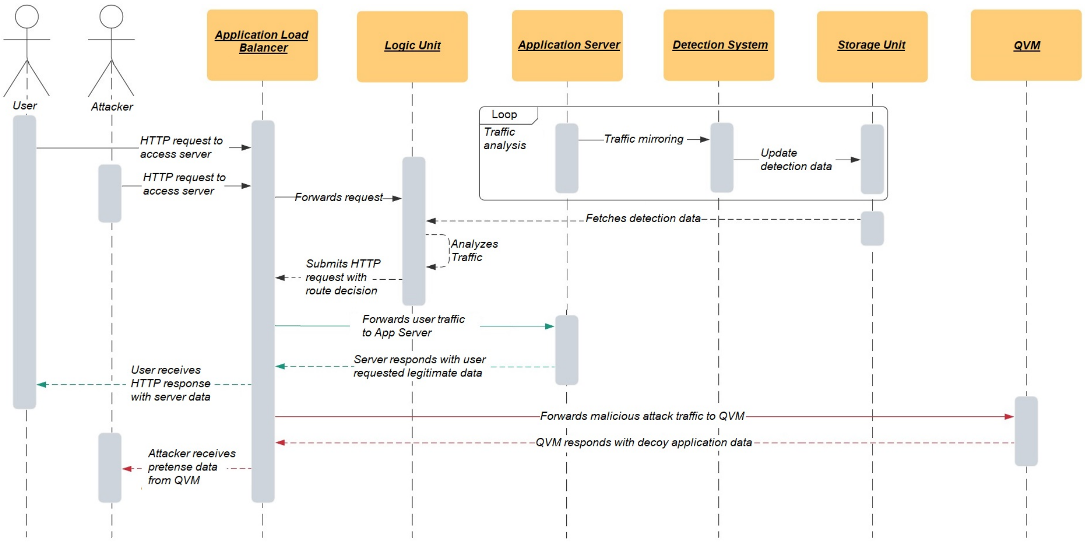
User
Attacker
Application Load Balancer
Logic Unit
Application Server
Detection System
Storage Unit
QVM
Loop
Traffic analysis
Traffic mirroring
Update detection data
HTTP request to access server
HTTP request to access server
Forwards request
Fetches detection data
Analyzes Traffic
Submits HTTP request with route decision
Forwards user traffic to App Server
Server responds with user requested legitimate data
User receives HTTP response with server data
Forwards malicious attack traffic to QVM
QVM responds with decoy application data
Attacker receives pretense data from QVM

Figure 4.4: Sequence diagram of OHDSI-Dolus defense system interactions for network traffic analysis and attack detection, along with attacker quarantine and active defense through initiation and maintenance of pretense.

Firstly, a user from internet accessing the cloud-based healthcare data applications makes requests to the listener based on rules configured in OHDSI. The listener then redirects the user's traffic to the target application server. An IDS placed inside the VPC is used detect network intrusions by sniffing network traffic flows in real-time to the OHDSI application server. If network intrusion or malicious activity is detected, the IDS will alert the system administrator, and then the adversary's IP address will be blocked at the application server. After blocking attackers IP address, the listener on the load balancer automatically redirect the traffic from the attacker to a Quarantine Virtual Machine (QVM) by using configured rules. Finally, by initiating pretense, the attacker will be deceived by being presented with decoy files of protected data to give a false sense of success.

Chapter 5

Active Defense Evaluation Results

In this section, we demonstrate the effectiveness of our two noval active defense frameworks. First, a rule-based 3QS-adaptation framework using tesbed setup that involves a virtual reality based interactive video learning platform viz. vSocial and defense by pretense framework data driven healthcare application viz. OHDSI-Dolus system on Amazon Web Services (AWS) resources. Our validation results show that our rule-based 3QS-adaptation framework's adaptation choices are effective in reducing the cybersickness levels and in maintaining the application functionality at a usable level. Further We evaluate our second active defense based *OHDSI-Dolus* system design by creating an experimental AWS testbed hosting a realistic OHDSI setup for protected health data analytics with electronic health record data (SynPUF) and publications data (CORD-19) related to COVID-19 [37]. Our experiment results show how we are able to successfully detect targeted attacks such as e.g., DDoS and create redirection of attack sources to QVMs. As a response from QVM, we successfully initiate a defense by pretense by sending fake HTTP responses and honey files from the decoy application to attackers mimicking the OHDSI application, which creates a false sense of success for the attackers.

5.1 Performance Evaluation of Rule-based 3QS Adaptation Framework

We setup our experimental testbed in a public cloud i.e., Amazon Web Services (AWS) [39] as shown in Figure 5.1. In this testbed, we host the open-source vSocial application [38] on an Amazon Elastic Compute Cloud (EC2) instance [39] to render the VRLE content to the users. We also host a controller node on another EC2 instance to: (i) capture network data using Amazon CloudWatch [39], and (ii) monitor the network data using our anomaly monitoring tool alongside a decision module hosted on a separate Jupyter notebook instance [39]. In addition, we store the captured and processed network data in the controller node into a DynamoDB [39] service. This DynamoDB service serves as a knowledge base for future anomaly events. We also connect our knowledge base to Amazon S3 [39] service using the Amazon Lambda [39] service in order to provide seamless interaction between the decision module and the anomaly monitoring tool. Before illustrating our experimental scenarios, we first detail the tools used for anomaly data collection required for our framework.

As part of anomaly data collection, we simulate a QoS issue (packet drop), QoA issue (packet drop + network lag), Security issue (DoS, packet duplication + packet tampering) in our vSocial application setup. We calculate the packet rate by capturing the raw data associated to the timestamp of each packet for each of the simulated 3QS issues along with the baseline data (of benign behavior) of the vSocial application. To simulate a DoS attack on vSocial, we used Clumsy 0.2 [40], a windows based tool to control networking conditions such as lag, drop, throttle, or tamper of live packets. To see the impact on our VRLE application performance, we specifically drop a certain percentage of live network packets. Using the Wireshark [41] tool, we capture packets being sent to-and-from our VRLE server in order to demonstrate possible data loss resulting from the packet capture. With the above specified tools and the experimental testbed setup, we

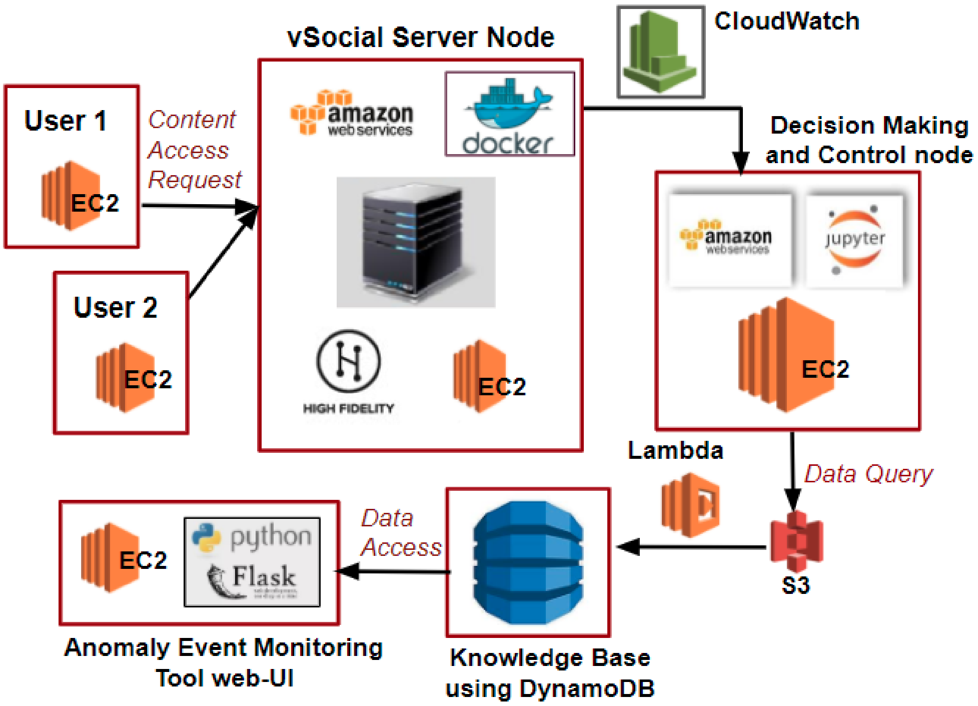

Figure 5.1: Experimental testbed setup for 3QS-adaptation framework evaluation using anomaly event monitoring and rule-based decision making.

collect the anomaly data relevant to 3QS issues in VRLE sessions.

To identify traces of 3QS anomaly events in the collected network data during a VRLE session, we developed a web-based anomaly monitoring tool using the Flask micro framework with Python3 [42]. Our anomaly monitoring tool uses AWS CloudWatch alarms to create triggers based on a threshold condition for every 3QS anomaly. For instance, a QoS alarm is triggered if the threshold condition *if ([No. of packets out] < 7280 packets/second)* fails. Similarly, for a QoA alarm, we use a threshold condition if the (CPU Utilization %) > 8%. Next, the anomaly monitoring tool will pass the collected anomaly data to the decision module of our 3QS-adaptation framework. We store this detected anomaly data into a AWS S3 bucket [39], which is further interfaced with DynamoDB [39], the knowledge base.

5.1.1 Adaptation Decision Making and Control Unit

With the captured anomaly data, the decision module will look up for the relevant adaptation and control module makes a decision to implement relevant adaption.

Table 5.1: Potential adaptation choices for different 3QS anomaly events.

Anomaly Issue	**Specific Category**	**Adaptation Name**
QoA	High CPU Utilization	Upgrading Instance Type (A1)
		Higher Resources (A2)
		Modifying Instance Volume (A3)
QoS	Low Network Bandwidth	Enabling Enhanced Networking (A4)
		Higher Network Bandwidth (A5)
Security	Denial of Service	Amazon Route 53 (A6)
		AWS GuardDuty (A7)
Intrusion	Unauthorized Access	Blacklist IP via third-party app (A8)

A sample list of potential adaptations for a specific decision unit are shown in the Table 5.1. For example, a QoA issue arising due to {packet drop + lag} can be mitigated using the adaptations in Table 5.1. Using the decision outcome, next the control module implements the adaptation based on the risk and cost aware analysis. For instance, for a security issue, we utilize the adaptation Blacklist IP (A8) to block unauthorized access based on the threshold condition (number of login attempts > 5). when an AWS alarm relevant to a security anomaly is triggered, the control module invokes an action for the suitable adaptation $A8$ keeping in mind the decision outcome, risk and cost factors. Similarly Based on such implementations for anomaly events in VRLE, we show the results of our adaptations using the “performance metrics" {response time, Threshold measures}, cost incurred in Table 5.2.

Once the decision related to an anomaly event is incorporated, its relevant information is updated into the knowledge base to train for future anomaly events. In our 3QS-adaptation framework, a knowledge base has been created using a DynamoDB service. To facilitate periodic updates from each of the modules in our framework into DynamoDB, we use the Amazon S3 service along with AWS Lambda functions. We use these both storage systems as our decision module that is hosted on a Jupyter notebook instance that takes only CSV data as input. The full capability of our knowledge base can be extended to other applications and can be utilized for employing additional adaptations in VRLE systems.

5.1.2 Quantification of Cybersickness for 3QS Anomalies

In this section, we objectively measure the induced cybersickness level for a given set of anomaly events i.e., visualization delay due to network lag (QoA issue), packet loss (QoS issue), and DoS attack (security attack). The works in [47] study that the quantifying effects of latency as the objective parameter to assess cybersickness. Based on the findings of the study, we measure the latency as the primary objective metric of cybersickness for several 3QS anomaly events in VRLE. Each of these attack anomaly events are simulated in different network conditions as detailed in our prior work [5]. We also found that 23.5 ms is the baseline latency for a normal functioning VRLE session, beyond which a user experiences cybersickness.

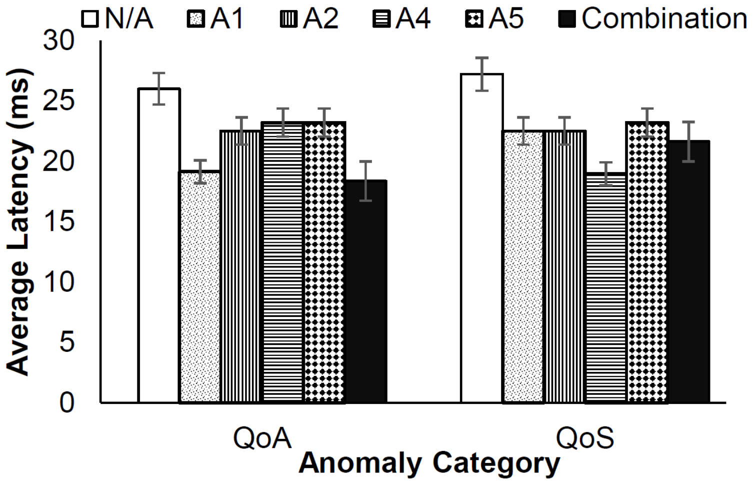

Figure 5.2: Avg. Latency measured (in ms) for QoA anomaly, QoS anomaly scenario in different adaptation scenarios.

The graphical results in Figure 5.2 detail the control of cybersickness level using latency metric for the adaptations (i.e., upgrading instance (A1), scaling of higher Resources (A2)) listed in Table 5.1. We also consider a no-adaptation (NA) scenario to study the adverse impact on cybersickness if no action is taken to control the raised anomaly event. Moreover, in real-world applications such as vSocial, there is a possibility that one adaptation action might not be enough to mitigate the anomaly impact, and an adaptation should consider the possibility of a combination of performance and security issues inducing cybersickness [7].

To address such an case, from the results in Figure 5.2, we observe that for a QoA anomaly, adaptations A1, A2 reduce the cybersickness by 26.43% and 13.46% respectively. In case of a QoS anomaly, the adaptation A4 reduces the cybersickness significantly by 30.28%. In addition, A1 and A2 reduce cybersickness by 17.28% making them the next suitable choice for a QoS anomaly as shown in Figure 5.2. We also note that the combination of best adaptations i.e., A1 and A4 reduces cybersickness by 29.39% for a QoA anomaly and 20.48% for a QoS anomaly event as shown in Figure 5.2. However the choice of combination can vary based on the considered list of potential candidates that can further impact the control of cybersickness levels in a VRLE session.

5.1.3 Risk and Cost Aware Trade-off Analysis

We term risk as "failure risk" which is a likelihood value of an adaptation that can fail in controlling the cybersickness for a given anomaly event. We adopt the NIST SP800-30 [43] based risk assessment method [48] where we use $L(D)$– the likelihood of decision of a specific adaptation and I represents the Impact of an adaptation in controlling the cybersickness level. We estimate the $L(D)$ based on the order of decision metrics. Using these both $L(D)$ and I, we calculate the failure risk as $R_f = 1 - f(L(D), I)$ where, $f(L(D), I)$ is the average function adopted from existing works [48]. We use a pre-defined semi-quantitative scale of 0-1 as guided by NIST for the impact/likelihood event assessments, with 1 indicating very high, and 0 indicating very low levels of impact. Using the latency measurement results in Figure 5.2, we consider the best/worst combination of adaptation choices for each anomaly event as illustrated in Figure 5.3.

We measure the performance metrics and system response time due to these adaptations using CloudWatch as shown in Table 5.2. With this, we highlight the functionality of our framework that takes dynamic decisions to control the cybersickness and maintain satisfactory application functionality. Based on our experimental results (i.e., cost-performance and risk evaluation), we enlist suitable

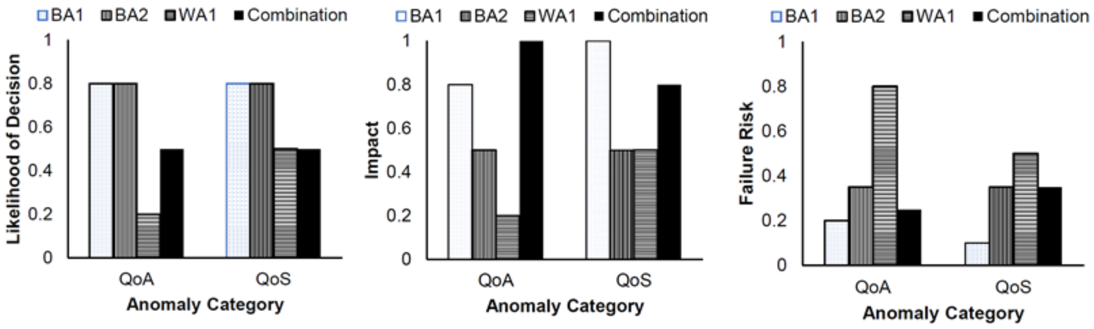

Figure 5.3: Risk evaluation associated with the best (BA1, BA2), worst (WA1) and combination of adaptations in controlling cybersickness for the given QoA and QoS anomaly event.

Table 5.2: Cost-aware application performance analysis of adaptations chosen for 3QS anomaly events.

Anomaly Event	**Adaptation name**	**Cost (in \$/hr)**	**Threshold Metric**	R_{at} **(in seconds)**
QoA	*A1*	0.23	CPU utilization rate is decreased to 4%	0.54
	A2	2.4		300
QoS	*A4*	0.10	Packet rate at 7280 packets/second	1
	A5	0.10		300
DoS	*A7*	0.33	Packet data measure	0.51
Unauthorized access	*A8*	0.02	Number of login attempts <5	Varies based on number of users

rules (i.e., best practices) to adopt for future anomaly events.

5.1.4 Recommendations Based on Key Findings

Based on our experimental evaluation of our framework to control cybersickness level using the listed adaptations in Table 5.1, we recommend rule-based practices as shown in Table 5.3. These practices are expressed in a semantic form i.e., we enlist Event-Condition-Action (ECA) rules with a typical form of $IF - THEN - (ELSE)$ [49] to adopt for future VRLE systems. From the results shown in Table 5.3, for a QoA anomaly with the given scenario in VRLE, we recommend adaptation A1 due to the low cost incurred and high impact on cybersickness control when compared to adaptation A2. Similarly, for an Unauthorized Access (UA), we recommend adaptation A8 over A7 due to the incurred cost and also the lack of control with the GuardDuty service in A7 as shown in

Table 5.3.

Table 5.3: Recommendations based on risk level (R_l), Cost level (C_l), and control on cybersickness (ΔCS).

IF		**THEN**				**ELSE**			
Anomaly	**Scenario in VRLE session**	A_i	R_l	C_l	$\Delta CS\%$	A_i	R_l	C_l	$\Delta CS\%$
QoA	Increasing number of users; To improve application run time	A1	L	L	26.43%	A2	M	M	13.46%
QoS	Lower latency in VRLE content	A4	L	L	30.28%	A1+A4	L	M	20.48%
UA	Only valid users in VRLE session	A8	L	L	20.7%	A7	M	H	-
DoS	Avoid loss of content availability	A1+A6	M	M	36.1%	A1+A7	M	H	-

In addition, our recommendations can range from ideas of checking for malware and updating security groups to extreme actions such as terminating the application instance altogether. Using such rule-based adaptations, we showcase the benefit of our proposed framework that controls the cybersickness level induced by the 3QS related anomalies.

5.2 Performance Evaluation of defense by pretense based OHDSI-Dolus system

Our OHDSI on AWS testbed set up is shown in Figure 5.4. There are three servers, including the application (OHDSI Server), the network-based IDS, and the QVM. All of these servers are set up using EC2 virtual instances, and are configured within the same virtual private cloud as private nodes, i.e., only accessible from other nodes within the testbed. While the data from the application and the QVM can be accessed by the users (benign or malicious) depending on their request type, the network-based IDS server is used solely for network traffic mirroring and for analysis involving attack detection. Our testbed also includes an Application Load Balancer (ALB), the only public-facing component in the testbed. The ALB works as a wrapper component which performs both the logic check and acts as a distributor depending on the load it receives, through which users can access the private servers by using their public IP addresses.

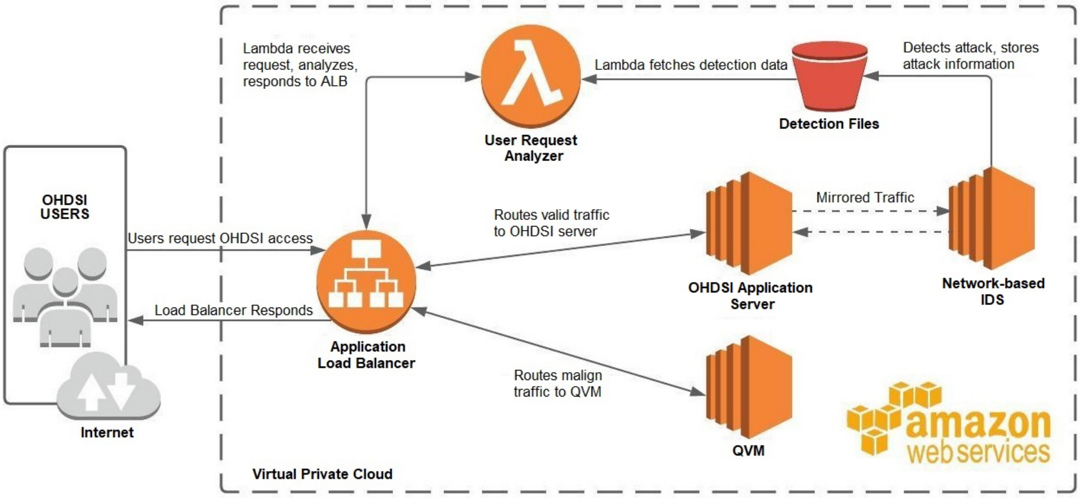
Lambda receives request, analyzes, responds to ALB
User Request Analyzer
Lambda fetches detection data
Detection Files
Detects attack, stores attack information
OHDSI USERS
Users request OHDSI access
Load Balancer Responds
Internet
Application Load Balancer
Routes valid traffic to OHDSI server
OHDSI Application Server
Mirrored Traffic
Network-based IDS
Routes malign traffic to QVM
QVM
amazon web services
Virtual Private Cloud

Figure 5.4: AWS testbed used to evaluate OHDSI-Dolus for targeted attacks.

The ALB has three listener rules, each pointing to a target server or AWS service and has a different priority. The first listener rule points to a Lambda function as a target which has the highest priority of all the rules. This Lambda function receives all the HTTP requests when users try to access the OHDSI application server, and will obtain the source IP of the user trying to access the server. Lambda function also fetches the blacklisted IP list from the AWS S3 bucket to match with the previously blacklisted attacker's IP addresses. The Lambda function then maps the source IP of the user with this list to conclude whether or not the traffic that the ALB is receiving is coming from an attacker IP. Once such a conclusion is made, the Lambda function responds to the ALB with an HTTP response that provides the re-route information which consists of the port numbers that the ALB uses to forward the traffic to either the OHDSI application server or to the QVM.

After receiving response from the Lambda function, the ALB will re-route traffic to the respective OHDSI application server or to the QVM. The other two listener rules point to the application server and the QVM. There is no need to prioritize these rules because both of them use different conditions, which will make them both exclusive of each other. For evaluation purposes, we use different ports to determine which server to re-route when a certain rule is matched.

The network-based IDS server constantly monitors the traffic for the OHDSI application server that has been mirrored to the network-based IDS using VPC traffic mirroring. It checks to categorize if an attack has occurred based on the network connection patterns and our attack detection logic. Upon detection of an attack, the network-based IDS server creates a list of IPs to be blacklisted and appends them to a database to keep track of the IPs that need to be re-directed to the QVM whenever a request is made from related IPs.

5.2.1 Risk Assessment Results

We evaluate the risks of various threat events in the STRIDE categories based on the NIST guidelines [43]. As shown in Table 5.4, the risk levels for different threats against the OHDSI healthcare application varies in the STRIDE model due to their distinct potential impact and likelihood values. The risk value of Data Alteration event under the Tampering category is the lowest among all the threats in the STRIDE categories. Due to the fact that the likelihood of tampering is relatively low (score of 3), which leads to a minimal chance of occurrence. Recently many techniques have been developed to enforce encryption of Data-at-Rest and Data-in-Transit. In addition, file integrity monitoring systems can be placed to deal with tampering threats, which also contributes to the low likelihood value in this category.

Table 5.4: Threat events related to OHDSI application with NIST-based guideline [43] used for risk calculation.

Category	Threat Events	Application Impact	STRIDE Threat	Likelihood	Impact	Threat Risk
A	IP spoofing	An attacker can alter the IP packet to gain access to Healthcare application server as authorised user.	Spoofing	5	21	Moderate (5)
B	Data Alteration	Malicious user can spoof the query to retrieve unauthorized data.	Tampering	3	10	Low (3)
C	Man-In-Middle attack	Attacker can impersonates as a user to retrieve unauthorized data.	Repudiation	5	6	Moderate (4)
D	SQL injection/Malware infection	An attacker can perform an SQL injection attack to affect the database or to gain access to unauthorized data. Also, the presence of malware on the system can lead to leakage of users' unauthorized data.	Information Disclosure	9	28	Very High (9)
E	DDoS Attack	Attacker can perform multiple SQL queries to overwhelm the database system.	Denial of Service	8	26	High (8)
F	Data Tampering	An attacker can tamper data because there's no integrity protection for data on the network.	Elevation of Privilege	3	24	Moderate (4)

On the other hand, the SQL injection/Malware infection under the Information Disclosure category and the DDoS attack under the Denial of Service are the two

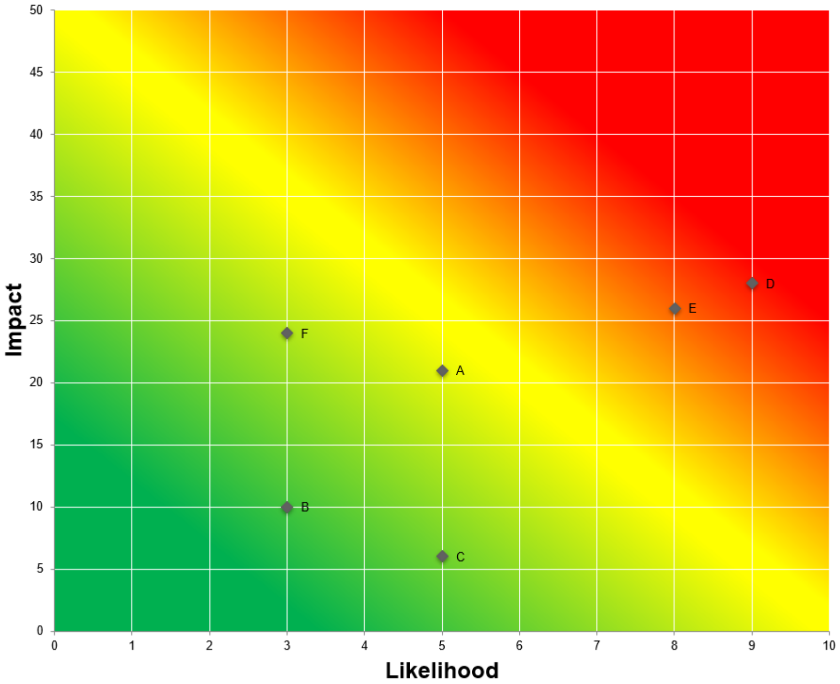

Figure 5.5: Heat map visualization of risk levels for different STRIDE categories A - F.

highest risk categories, with risk levels of 9 and 8, respectively. This is due to the fact that the impact and likelihood are both high in these two categories relating to modern healthcare data processing applications, as opposed to Data Tampering, which also has a high impact but a very low likelihood of occurrence. The rest of the STRIDE categories have risk values that lie in between the Tampering and Information Disclosure categories, indicating that they represent moderate threats in the healthcare data processing pipeline applications. The risk levels are also visualized in the heat map in Figure 5.5, where the red color represents high risk, green color represents low risk, and yellow color represents medium risk in the relative STRIDE categories A - F.

5.2.2 Detection Results

We present the performance of our OHDSI-Dolus considering an exemplar attack with a high risk level, i.e., the DDoS attack, which is one of the most prevalent attacks in the STRIDE categories with regarding to OHDSI health data processing applications. Recall that the QoD parameter determines the overall the accuracy and speed of cyber-attack detection impacting the OHDSI healthcare application. We evaluate the QoD metric based on the ensemble learning accuracy and time taken for the models to run in OHDSI-Dolus, and compare the performance with state-of-the-art detection schemes in [30] and [44].

To identify potential network intrusion or a cyber-attack event, we setup our network-based IDS in our VPC on an EC2 instance. We use the ensemble learning based detection scheme used as part of the OHDSI-Dolus related IDS implementation. We take advantage of the AWS VPC traffic mirroring service to mirror the network traffic flowing into our VPC that is routed to the IDS. We also used the AWS Cloud Watch service to monitor the OHDSI application server's network flow i.e., mirrored network traffic, as shown in Figure 5.6. We can see from the graph, the different levels of network packets mirrored from the OHDSI server during the EC2 instance initiation, OHDSI application server launch and during user data query.

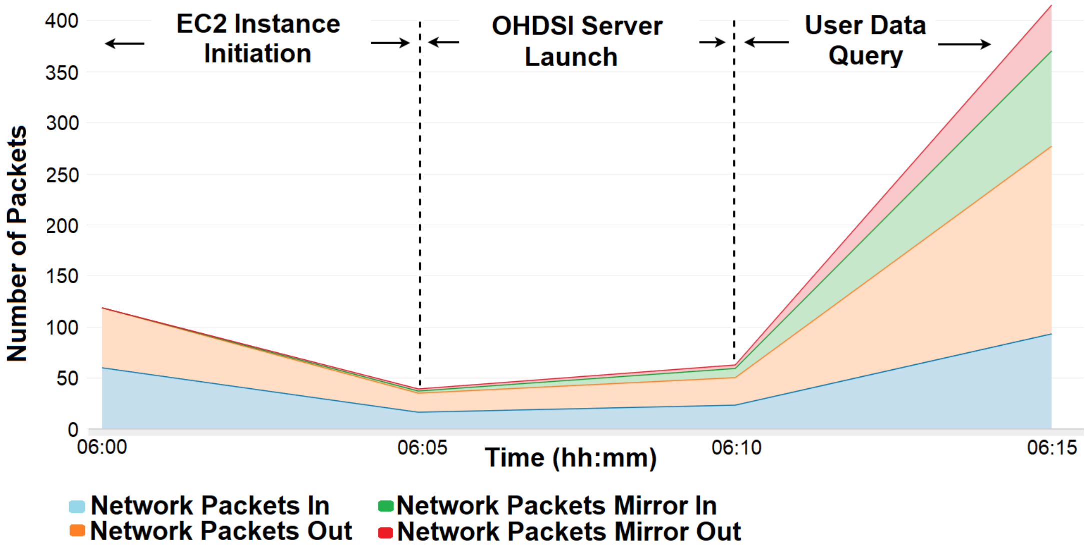

Figure 5.6: Dolus mirroring OHDSI server traffic at different stages for analysis and detection as viewed with AWS Cloud Watch.

To evaluate the performance of our OHDSI-Dolus mechanism, we compare its performance with other DDoS detection mechanisms presented in [30] and [44]. We primarily choose these works for comparison since the studies presented take advantage of ML based attack detection against cloud-based systems. The work in [30] presents CS_DDoS, a framework for detection and prevention of DDoS attack in cloud environments. In CS_DDoS, the incoming packets are classified using several machine learning models to decide whether the sources are associated with a genuine client or an attacker based on feature matching.

Authors in [44] propose to use the extreme gradient boosting (XGBoost) as the detection method in a cloud platform with software-defined networking. The detection results validate that XGBoost performs relatively better than CS_DDoS with higher accuracy, lower false positive rate, fast-speed and has scalability in detection of DDoS attacks.

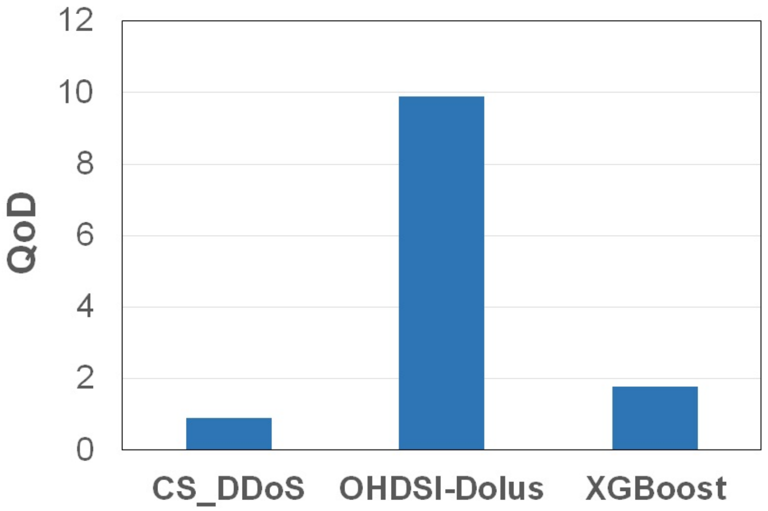

Figure 5.7: Quality of Detection (QoD) results of DDoS attack detection based on accuracy and time taken by state-of-the-art detection mechanisms in comparison with our OHDSI-Dolus.

Figure 5.7 shows that our healthcare data processing pipeline, the OHDSI, equipped with OHDSI-Dolus outperforms the state-of-the-art mechanisms viz., CS_DDoS and XGBoost in DDoS attack detection. The comparison is done using the accuracy over time QoD calculation in Equation (4.1). In the results, we average the accuracy of different machine learning models used across the average of time they took for the calculation for QoD.

5.2.3 Defense by Pretense Qualitative Evaluation

Upon detection of a DDoS attack, our OHDSI-Dolus system initiates attack mitigation by re-routing the network traffic from the attacker to the QVM. In the QVM, we successfully deployed a decoy service that mimics a running OHDSI application server. This server serves dummy honey files that are intended only to be provided to attackers to maintain the pretense, and the attackers are led to believe they have gained access to the main OHDSI application server, while in reality they have been deceived.

We perform qualitative comparison between our OHDSI-Dolus system with the defense scheme presented in the works of [30] and [31]. The CS_DDoS framework in [30] uses IP blacklisting to mitigate DDoS attacks on a cloud platform. In contrast, [31] proposes a mechanism to mitigate data ex-filtration attacks using deception in cloud platforms. Whenever a download or sharing request is made, if the host MAC address does not match the embedded identifier within the file, the corresponding decoy document (instead of the actual file) is returned to deceive the attacker.

Table 5.5 summarizes the main features of our OHDSI-Dolus and qualitatively compares it with two state-of-the-art mechanisms i.e., CS_DDoS [30] and Deception based defense [31] in terms of the features, advantages, use case spectrum, and their limitations. We show that our OHDSI-Dolus scheme is more scalable, cost-effective, and easier to deploy as we take advantage of low-cost and most commonly used services provided by public cloud providers. Thereby, using our OHDSI-Dolus, the attackers can be effectively engaged with a QVM that helps to

Table 5.5: Comparison of OHDSI-Dolus performance with state-of-the-art active defense mechanisms that have the potential to be used for protection of cloud-based healthcare data processing pipelines.

	CS_DDoS Detection/Defense [30]	**Deception based Defense [31]**	**OHDSI-Dolus Detection/Defense**
Features	ML-based detection attacks and prevention using IP blacklisting.	Detection using detection engine and prevention by generating decoy documents.	Detection of cyber-attacks by ML-enabled network-based IDS and mitigation by initiating Pretense.
Advantages	Able to reduce bandwidth consumption by early detection.	Detection without relying on cloud providers using cyber deception.	Scalable, Cost-effective and easy to deploy.
Suitable for	Early DDoS attack detection.	Generation of decoy objects based on input.	The scalable and cost-effective mechanism for early detection and mitigation.
Limitations	Detection fails if attacker is using spoofed IP's.	Detection scheme can be spoofed by spoofing MAC addresses.	The proposed scheme is based on relevant services provided by cloud providers.

gain more threat intelligence information on the attacker and the corresponding attack vectors.

Chapter 6

Conclusion

Active defense is a state-of-the-art paradigm where proactive or reactive cybersecurity strategies are used to augment passive defense policies (e.g., firewalls). It involves using knowledge of the adversary to create of dynamic policy measures to secure resources and outsmart adversaries to make cyber-attacks difficult to execute. In this work, We developed two novel active defense strategies to mitigate the impact of security anomaly events within: (a) novel Rule-Based Performance and Security (3QS) Adaptation Framework to mitigate the impact of performance and security anomaly events that induce cybersickness in social virtual reality learning environment (VRLE), and (b) a novel cloud-based attack detection and active defense mechanism viz., "OHDSI-Dolus" for a cloud-hosted healthcare data sharing environment (HDSE). In our rule based adaptaion framework We quantified the cybersickness metric objectively using a latency metric for a simulated anomaly event scenario. We utilized a priority-based queuing model that handles anomaly events in the order of highest cybersickness inducing levels. To determine the suitable adaptation for handling a given anomaly event type, our approach involves performing risk and cost aware analysis for each decision outcome. Once a suitable adaptation is incorporated for a given anomaly event type,cybersickness measurements are updated and used as feedback to determine the impact on the anomaly event. Our validation results for rule based adaptaion framework show that the real-time adaptations suggested by our rule-based framework: (i) reduce the cybersickness level by 26.43% for a QoA anomaly and the same for a QoS anomaly event by 30.28%, and (ii) maintains the application functionality within the threshold limit (beyond which an application is non-functional) along with low system response times. Based on these key findings, we enlisted suitable practices for prevention of 3QS issues based on NIST SP800-160 guidelines. Furthermore, in our cloud-based attack detection and active defense mechanism we analyzed unique attack surfaces in the healthcare data processing pipelines using the Microsoft STRIDE methodology and performed a related risk assessment based on the NIST guidelines to identify the prominent threats such as DDoS attack and APT. Based on the the risk assessment,we developed a design for an active defense solution i.e., OHDSI-Dolus that can be integrated with healthcare data processing pipelines. Moreover, we showed that through active defense strategies, our OHDSI-Dolus system is capable of threat detection and provides threat mitigation services to effectively defend against targeted attacks in a robust manner.We showed how our OHDSI-Dolus system actually takes advantage of "defense by pretense" theory for mitigation of threats such as DDoS and APTs for cloud-based healthcare data processing pipelines by luring the attacker to quarantine virtual machine instances.

Bibliography

[1] De Gaspari, F., Jajodia, S., Mancini, L.V. and Panico, A., 2016, October. Ahead: A new architecture for active defense. In *Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense* (pp. 11-16).

[2] Mairh, A., Barik, D., Verma, K. and Jena, D., 2011, February. Honeypot in network security: a survey. In *Proceedings of the 2011 international conference on communication, computing security* (pp. 600-605).

[3] Dai, N.H.P., András, K. and Zoltán, R., 2016. E-learning security risks and counter measures. *Engineering research and solutions in ICT, 1,* pp.17-25.

[4] Zizza, C., Starr, A., Hudson, D., Nuguri, S.S., Calyam, P. and He, Z., 2018, January. Towards a social virtual reality learning environment in high fidelity. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)* (pp. 1-4). IEEE.

[5] Gulhane, A., Vyas, A., Mitra, R., Oruche, R., Hoefer, G., Valluripally, S., Calyam, P. and Hoque, K.A., 2019, January. Security, privacy and safety risk assessment for virtual reality learning environment applications. In *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)* (pp. 1-9). IEEE.

[6] Jia, J. and Chen, W., 2017, July. The ethical dilemmas of virtual reality application in entertainment. In *2017 IEEE International Conference on Computa-*

tional Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (Vol. 1, pp. 696-699). IEEE.

[7] Valluripally, S., Gulhane, A., Mitra, R., Hoque, K.A. and Calyam, P., 2020, January. Attack trees for security and privacy in social virtual reality learning environments. In 2*2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)* (pp. 1-9). IEEE.

[8] Wang, S., Valluripally, S., Mitra, R., Nuguri, S.S., Salah, K. and Calyam, P., 2019, June. Cost-performance trade-offs in fog computing for IoT data processing of social virtual reality. In *2019 IEEE International Conference on Fog Computing (ICFC)* (pp. 134-143). IEEE.

[9] Casey, P., Baggili, I. and Yarramreddy, A., 2019. Immersive virtual reality attacks and the human joystick. *IEEE Transactions on Dependable and Secure Computing, 18(2),* pp.550-562.

[10] Ghahramani, M.H., Zhou, M. and Hon, C.T., 2017. Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services. *IEEE/CAA Journal of Automatica Sinica, 4(1),* pp.6-18.

[11] B. Mukherjee, R. L. Neupane, and P. Calyam, "End-to-end iot security middleware for cloud-fog communication," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud).IEEE, 2017,* pp. 151–156

[12] Rebenitsch, L. and Owen, C., 2016. Review on cybersickness in applications and visual displays. *Virtual Reality*, 20(2), pp.101-125.

[13] LaViola Jr, J.J., 2000. A discussion of cybersickness in virtual environments. *ACM Sigchi Bulletin*, 32(1), pp.47-56.

[14] Kuo, M.H., 2011. Opportunities and challenges of cloud computing to improve health care services. *Journal of medical Internet research,* 13(3), p.e67.

[15] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control,"*Journal of medical systems,* vol. 40, no. 10, pp. 1–8, 2016

[16] Abaid, Z., Shaghaghi, A., Gunawardena, R., Seneviratne, S., Seneviratne, A. and Jha, S., 2019, May. Health access broker: Secure, patient-controlled management of personal health records in the cloud. *In Computational Intelligence in Security for Information Systems Conference* (pp. 111-121). Springer, Cham.

[17] Abouelmehdi, K., Beni-Hessane, A. and Khaloufi, H., 2018. Big healthcare data: preserving security and privacy. *Journal of Big Data,* 5(1), pp.1-18.

[18] Khorshed, M.T., Ali, A.S. and Wasimi, S.A., 2012. A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing. *Future Generation computer systems,* 28(6), pp.833-851.

[19] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review,"*Journal of Network and Computer Applications,* vol. 36, no. 1, pp. 16–24, 2013

[20] Kanimozhi, V. and Jacob, T.P., 2019, April. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In *2019 international conference on communication and signal processing (ICCSP)* (pp. 0033-0036). IEEE.

[21] Aborujilah, A. and Musa, S., 2017. Cloud-based DDoS HTTP attack detection using covariance matrix approach. *Journal of Computer Networks and Communications,* 2017.

[22] He, Z., Zhang, T. and Lee, R.B., 2017, June. Machine learning based DDoS attack detection from source side in cloud. In *2017 IEEE 4th International*

Conference on Cyber Security and Cloud Computing (CSCloud) (pp. 114-120). IEEE.

[23] Choi, J., Choi, C., Ko, B. and Kim, P., 2014. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. *Soft Computing,* 18(9), pp.1697-1703.

[24] Kim, J., Lee, T., Kim, H.G. and Park, H., 2013. Detection of advanced persistent threat by analyzing the big data log. *Advanced Science and Technology Letters,* 29, pp.30-36.

[25] Bhatt, P., Yano, E.T. and Gustavsson, P., 2014, April. Towards a framework to detect multi-stage advanced persistent threats attacks. In *2014 IEEE 8th international symposium on service oriented system engineering* (pp. 390-395). IEEE.

[26] Binde, B., McRee, R. and O'Connor, T.J., 2011. Assessing outbound traffic to uncover advanced persistent threat. *SANS Institute.* Whitepaper, 16.

[27] Vukalović, J. and Delija, D., 2015, May. Advanced persistent threats-detection and defense. In *2015 38Th international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1324-1330). IEEE.

[28] Stout, W., Urias, V., Loverro, C. and Anthony, B., 2017. Now You See Me Now You Don't: Advancing Network Defense through Network Deception (No. SAND2017-8892C). *Sandia National Lab.(SNL-NM),* Albuquerque, NM (United States).

[29] Bringer, M.L., Chelmecki, C.A. and Fujinoki, H., 2012. A survey: Recent advances and future trends in honeypot research. *International Journal of Computer Network and Information Security,* 4(10), p.63.

[30] A. Sahi, D. Lai, Y. Li, and M. Diykh, “An efficient ddos tcp flood attack-detection and prevention system in a cloud environment,”*IEEE Access,vol. 5,* pp. 6036–6048, 2017.

[31] Wilson, D. and Avery, J., 2016. Mitigating Data Exfiltration in Storage-as-a-Service Clouds. *arXiv preprint* arXiv:1606.08378.

[32] Araujo, F., Hamlen, K.W., Biedermann, S. and Katzenbeisser, S., 2014, November. From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security* (pp. 942-953).

[33] Cifranic, N., Hallman, R.A., Romero-Mariona, J., Souza, B., Calton, T. and Coca, G., 2020. Decepti-SCADA: A cyber deception framework for active defense of networked critical infrastructures. *Internet of Things,* 12, p.100320.

[34] Vassell, M., Apperson, O., Calyam, P., Gillis, J. and Ahmad, S., 2016, January. Intelligent Dashboard for augmented reality based incident command response co-ordination. In *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)* (pp. 976-979). IEEE.

[35] Neupane, R.L., Neely, T., Calyam, P., Chettri, N., Vassell, M. and Durairajan, R., 2019. Intelligent defense using pretense against targeted attacks in cloud platforms. *Future Generation Computer Systems,* 93, pp.609-626.

[36] (2011) S. Shekyan, SlowHTTPTest. Application Layer DoS attack.[Online]. Available: *https://github.com/shekyan/slowhttptest/wiki[33]*

[37] Alarcon, A.M.L., Oruche, R. and Calyam, P., Cloud-based data pipeline orchestration platform for covid-19 evidence-based analytics. Submitted for publication.

[38] Zizza, C., Starr, A., Hudson, D., Nuguri, S.S., Calyam, P. and He, Z., 2018, January. Towards a social virtual reality learning environment in high fidelity.

In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)* (pp. 1-4). IEEE.

[39] Amazon web services. Last accessed 2021-04-02. [Online]. *Available:https://aws.amazon.com/*

[40] "clumsy 0.2", 2018. Last accessed 2021-04-02. [Online]. *Available:https://jagt.github.io/clumsy*

[41] Wireshark tool. Last accessed 2021-04-02. [Online]. *Available:https://www.wireshark.org/*

[42] Flask: A web-development guide. Last accessed 2021-04-02. [Online]. *Available: https://pypi.org/project/Flask/*

[43] Ross, R.S., 2012. Guide for conducting risk assessments (nist sp-800-30rev1). *The National Institute of Standards and Technology (NIST), Gaithersburg.*

[44] Chen, Z., Jiang, F., Cheng, Y., Gu, X., Liu, W. and Peng, J., 2018, January. XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In *2018 IEEE international conference on big data and smart computing (bigcomp)* (pp. 251-256). IEEE.

[45] Rebenitsch, L. and Owen, C., 2016. Review on cybersickness in applications and visual displays. *Virtual Reality,* 20(2), pp.101-125.

[46] LaViola Jr, J.J., 2000. A discussion of cybersickness in virtual environments. *ACM Sigchi Bulletin,* 32(1), pp.47-56.

[47] G. Samaraweera, R. Guo, and J. Quarles, "Latency and avatars invirtual environments and the effects on gait for persons with mobilityimpairments," in *2013 IEEE Symposium on 3D User Interfaces (3DUI).IEEE, 2013,* pp. 23–30.

[48] Dickinson, M., Debroy, S., Calyam, P., Valluripally, S., Zhang, Y., Antequera, R.B., Joshi, T., White, T. and Xu, D., 2018. Multi-cloud performance and

security driven federated workflow management. *IEEE Transactions on Cloud Computing, 9(1),* pp.240-257.

[49] Müller, R., Greiner, U. and Rahm, E., 2004. Agentwork: a workflow system supporting rule-based workflow adaptation. *Data Knowledge Engineering,* 51(2), pp.223-256.

[50] Bhagwan, R. and Lin, B., 2000, March. Fast and scalable priority queue architecture for high-speed network switches. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)* (Vol. 2, pp. 538-547). IEEE.

[51] Hripcsak, G., Duke, J.D., Shah, N.H., Reich, C.G., Huser, V., Schuemie, M.J., Suchard, M.A., Park, R.W., Wong, I.C.K., Rijnbeek, P.R. and Van Der Lei, J., 2015. Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers. *Studies in health technology and informatics,* 216, p.574.

[52] Jiang, L., Chen, H. and Deng, F., 2010, May. A security evaluation method based on STRIDE model for web service. In *2010 2nd International Workshop on Intelligent Systems and Applications (pp. 1-5).* IEEE.

[53] You, D., Seo, B.S., Jeong, E. and Kim, D.H., 2018. Internet of Things (IoT) for seamless virtual reality space: Challenges and perspectives. *IEEE Access,* 6, pp.40439-40449.

[54] Abulrub, A.H.G., Attridge, A.N. and Williams, M.A., 2011, April. Virtual reality in engineering education: The future of creative learning. In *2011 IEEE global engineering education conference (EDUCON)* (pp. 751-757). IEEE.

[55] Allcoat, D. and von Mühlenen, A., 2018. Learning in virtual reality: Effects on performance, emotion and engagement. *Research in Learning Technology,* 26.

[56] Jonesi, S., Adams, J., Valluripally, S., Calyam, P., Hittle, B. and Lai, A., 2018, July. QOE Tuning for Remote Access of Interactive Volume Visualization Applications. In *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)* (pp. 1-6). IEEE.

[57] Chemodanov, D., Calyam, P., Valluripally, S., Trinh, H., Patman, J. and Palaniappan, K., 2018. On qoe-oriented cloud service orchestration for application providers. *IEEE Transactions on Services Computing.*

[58] Khoshkbarforoushha, A., Khosravian, A. and Ranjan, R., 2017. Elasticity management of streaming data analytics flows on clouds. *Journal of Computer and System Sciences,* 89, pp.24-40.

[59] Simmhan, Y., Cao, B., Giakkoupis, M. and Prasanna, V.K., 2011, June. Adaptive rate stream processing for smart grid applications on clouds. In *Proceedings of the 2nd international workshop on Scientific cloud computing* (pp. 33-38).

[60] Sukhov, A., Calyam, P., Daly, W. and Ilin, A., 2005. Towards an analytical model for characterizing behavior of high-speed VVoIP applications. *Computational Methods in Science and Technology,* 11(2), pp.161-167.

[61] Calyam, P., Rajagopalan, S., Selvadhurai, A., Mohan, S., Venkataraman, A., Berryman, A. and Ramnath, R., 2013, May. Leveraging OpenFlow for resource placement of virtual desktop cloud applications. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)* (pp. 311-319). IEEE.

[62] Hadim, S. and Mohamed, N., 2006. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE distributed systems online,* 7(3), pp.1-1.

[63] Valluripally, S., Akashe, V., Fisher, M., Falana, D., Hoque, K.A. and Calyam, P., 2021, August. Rule-based Adaptations to Control Cybersickness in Social

Virtual Reality Learning Environments. In *21 8th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 350-358).* IEEE.

[64] Akashe, V., Neupane, R.L., Alarcon, M.L., Wang, S. and Calyam, P., 2021, July. Network-based Active Defense for Securing Cloud-based Healthcare Data Processing Pipelines. In *2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-9).* IEEE.