

User Safety at Risk: Assessing Potential Threats Causing Cybersickness in Virtual Reality

Abstract—Sociotechnical systems such as Virtual Reality (VR) involves a complex user interaction among the components of the immersive environment. Application domains for this class of systems include disaster training, learning environments, flight simulations, military training and so on. Traditionally, such sociotechnical applications focus mainly on the provision of user experience. However, these applications ignore user safety as their primary concern, such as *cybersickness*. Our study shows that cyber-attacks on network connected components such as IoT devices in VR applications may induce cybersickness to the end users. To be more specific, in this paper, we propose a risk assessment methodology to identify and evaluate the potential threat scenarios that may cause cybersickness in VR environments. We develop novel threat models and formalize them using attack trees. Consequently, the developed attack trees are translated into the stochastic timed automata and then quantitative evaluation is performed using the statistical model checking technique for different attacker profiles. We illustrate the effectiveness of our approach on a VR Learning Environment (VRLE) application. Such an analysis can help to incorporate user safety as the main concern while designing distributed sociotechnical applications.

Index Terms—Virtual reality; attack trees; safety; cyber sickness; risk assessment; formal verification

I. INTRODUCTION

VR applications are the immersive environments where users interact with each other and the network connected components. A widespread adoption of sociotechnical functionalities in such environments emerges novel application areas such as: military, flight simulations, disaster training, education [1] [2] [3], medical field (therapy treatments) [4] [5]. With the advancement in VR technology, new capabilities to enhance user interactions and experience are developed. For instance, VR has the ability to transfer the real contexts into simulations as part of interactive learning in a multi-user scenario [6]. However, there are certain undesirable effects that disrupts *user safety* due to the exposure in virtual environment such as *cybersickness*, which is a set of unpleasant symptoms such as eyestrain, headache, nausea or even vomiting [7]. Existing literature discusses that inducing factors of cybersickness are mainly due to physiological conditions. But, other issues ranging from hardware to software components, design issues, potential threats that aim for cyber attacks in VR applications are unexplored on a large scale.

In other words, any security, privacy breach of the sensitive information which is collected and stored in such plethora of connected devices can put user safety in jeopardy. For instance, an attacker controlling and changing the orientation of avatar, intruding into the user's VR space, tampering the VR content

with spooky visuals may disrupt the user causing nausea or simulator sickness. In addition to that, several VR applications which are connected with a plethora of networked devices can be prone to intermittent network discrepancy attacks which causes disruptions during VR sessions, thus leading to cybersickness.

There exists several solution approaches that addresses both social, technical aspects to provide immersiveness such as [8] for an information security design application. However, such approaches fail to address user safety, which can be affected due to compromise of security, privacy issues when designing the application systems. Due to such discrepancies and overexposure in the socio-technical systems, the users are disrupted as discussed in [9]. The authors in [9], explains the causes of cybersickness which occurs in a VRLE developed for training youth with ASD. A seminal work on safety issues for virtual environments [10] establishes that the human performance efficiency is affected by task and user characteristics. Existing works such as [11], compared a virtual environment in a display monitor with head-mounted display (HMD) to establish its correlation with cybersickness. Similarly, literature reviews [12] highlights the aspects of VR technology that can cause symptoms like cybersickness and their consequences. In addition to this, existing social and technical methods are used to address the threats on user sensitive information to discusses about preliminary framework approaches ignoring the user safety as the priority [13] [14] [15]. Thus, with the growing nature of application domains in VR, it is imperative to address user safety as primary goal in developing new VR architecture design along with the social, technical, provision of user experience facets.

In this paper, we focus on the causes of cybersickness for a Virtual Reality Learning Environment (VRLE) case study: vSocial [16] which was developed for youth with autism spectrum disorder (ASD). The VR application case study consists many features that improve the VRLE application usability. However, it also generate new attack possibilities to disrupt user safety. Since every VR users in such applications are susceptible to cybersickness in a different manner [12], we refine our problem scope to the following research questions:

- What components (sensors, controllers, headsets, etc.,) in VR are vulnerable and what type of cyber-attacks on those identified components may lead to cybersickness?
- What are the VR applications issues (network, storage, etc.) that might also triggers cybersickness?

With this motivation, this paper proposes a methodology

that addresses the above mentioned research questions. To the knowledge of the authors, this is the first research that identifies and evaluates the potential threat factors (security, privacy) inducing cybersickness (user safety). Our proposed methodology develops a novel threat model using the attack tree formalism that captures the interplay of security and privacy facets on cybersickness in VR applications. To elucidate, our solution is based on stochastic model checking (SMC) approach, that gives the feasibility to build larger models by composing smaller ones [17]. Furthermore, SMC allows simulation of complex systems where a simple closed-form solution does not exist, or a rigorous state space search is infeasible, as in our VR case studies. With the capability of the compositional SMC approach, we translate the attack tree (composed with leaf, intermediate, root nodes and gates) into individual stochastic timed automata (STA), and use the state-of-the-art UPPAAL SMC model checker for the analysis. To be more specific, we perform quantitative evaluation of the translated STA model using the SMC queries and identify the most vulnerable threat components that induces cybersickness. With the obtained analysis, we also perform a risk assessment of the identified threat vectors on cybersickness. Thus, we evaluate the user safety in a networked VR setup by determining the most critical components and attack scenarios.

The outline of the paper is as follows: Section II presents the background of our proposed approach. Section III describes the proposed methodology in detail while Section IV outlines the evaluation methodology of our approach. Section V discusses results and some important aspects of our approach. Section VI concludes the work with directions of future research.

II. PRELIMINARIES

A. VR application case study: vSocial

To illustrate our problem scope, we consider a VRLE application: vSocial [16] as the case study. This VR application has several functionalities such as user access control, managing VR session permissions, session progress tracking, access control, network performance. In addition to this, to understand the user's progress and their experience in the environment, vSocial has the capability to collect Emotion (EEG) data of users using Muse headsets and display at the instructor's web application. The major components of the vSocial application includes: HMD devices (HTC Vive) that has a VR headset, two hand-held controllers, and two base stations for accurate localization and tracking of the controllers as shown in Figure 1. The headset facilitates the users to connect to the Virtual classroom and a VRLE cloud server on GENI through high-speed networks. Users interact with the environment, instructors and peers using their virtual hands (controllers) and voices. The cloud server delivers the learning content to the virtual classroom and stores student engagement data based on their participation in these classroom activities, all in real-time. The vSocial application includes: VR rendering, session permissions, web applications and instructional

contents hosted as web pages to provide an immersive learning environment. With such inherent capabilities to provide an immersive user experience, there are certain user safety discrepancies such as cybersickness. Failure to address these issues may result in disrupting users via some forms cyber-attack, for instance via: (i) insertion of malicious scripts in the instructional content causing flickering of the visuals, and (ii) controlling the environment via intrusion. In short, addressing user safety is of paramount importance for the adoption of sociotechnical VR applications as learning environments.

B. UPPAAL SMC

In this work, to analyze the attack trees we utilize the UPPAAL SMC model checker [18] that is widespread for modeling, validation and verification of real-time systems. The modeling language of UPPAAL offers additional features such as bounded variables to model the behaviour of the real-time systems. We convert the developed safety attack trees into stochastic timed automata which is then used for model checking using the UPPAAL SMC tool. We use the property specification language WMTL [19] to translate the cybersickness related metrics into UPPAAL SMC queries. For instance, if we indicate the goal state in the STA of *Top_event* as *Fail*, then the probability of a successful disruption within time t can be written as:

$$Pr[x \leq t](\Diamond Top_event.Fail) \quad (1)$$

where (\Diamond) and x is a clock in the STA to track the global time. Probability estimate of the query is given as a estimation interval $[Pr - \epsilon, Pr + \epsilon]$. The detail syntax and semantics of the SMC query language can be found in [20].

C. Stochastic timed automata

Stochastic timed automata (STA) is an extension of timed automata with stochastic semantics. Although attack trees can act as a fine model to find potential vulnerabilities and threats in the system, using STA for evaluation uncovers plethora of quantitative evaluation methods. Works in [17] describes an approach to use STA for attack tree analysis using UPPAAL SMC [18]. A STA associates logical locations with continuous, generally distributed sojourn times [21]. In STA, constraints on edges and invariants on locations, such as clocks are used to enable transition from one state to another [17].

Definition 1 (Stochastic Timed Automata). Given a timed automata which is equipped with assignment of invariants \mathcal{I} to locations \mathcal{L} , we formulate an STA as a tuple $T = \langle \mathcal{L}, l_{init}, \Sigma, \mathcal{X}, \mathcal{E}, \mathcal{I}, W \rangle$, where \mathcal{L} is a finite set of locations, $l_{init} \in \mathcal{L}$ is the initial location, Σ is a finite set of actions, \mathcal{X} is the finite set of clocks, $\mathcal{E} \subseteq \mathcal{L} \times \mathcal{L}_{clk} \times \Sigma \times 2^{\mathcal{X}}$ is a finite set of edges, with \mathcal{L}_{clk} representing the set of clock constraints, $\mathcal{I}: \mathcal{L} \rightarrow \lambda$ is the invariant where λ is the rate of exponential assigned to the locations \mathcal{L} , W is the weights specified over the dotted edges to specify as probability distribution for discrete transitions.

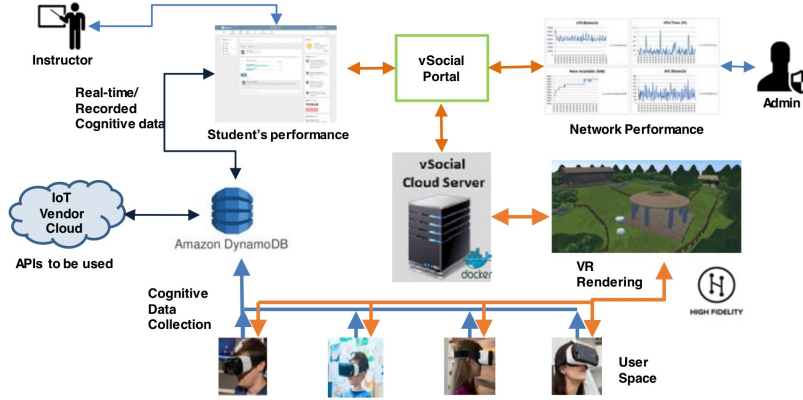


Figure 1: vSocial system components used for real-time student learning environment

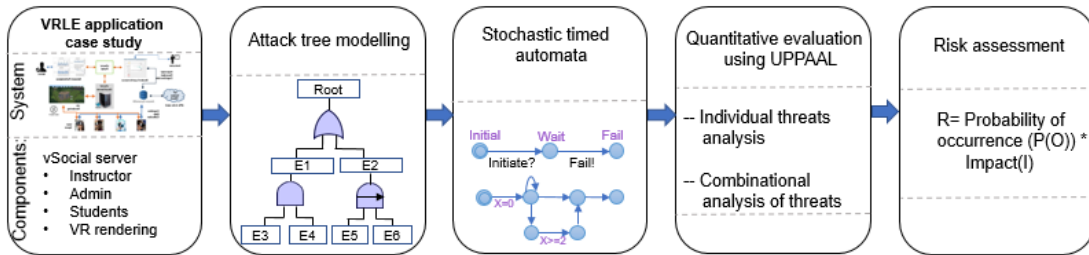


Figure 2: Framework for safty attack attack tree analysis of social virtual reality learning environment

III. PROPOSED METHODOLOGY

In this section, we present our proposed methodology that emphasizes on threat factors causing cybersickness for the users in VR applications. To elucidate, using our VR application case study, we model the potential attack vectors in the form of safety attack trees¹. Secondly, we perform statistical model checking on the STA's which are translated from the obtained safety attack tree. Thirdly, we showcase an in-depth analysis of the vulnerable threat factors due to application issues and VR architecture components causing cybersickness. Lastly, we perform risk assessment against the identified attack vectors to evaluate the dependability of the VR system. An overview of our proposed methodology is shown in Figure 2.

A. Formalization of attack trees

Attack trees are hierarchy based graphical models that show how the goal of attacker (root node) can be refined into smaller goals (intermediate nodes) until it reaches to the leaf nodes (BAS, child nodes) where no further refinement is possible.

Definition 2 (Attack trees). An attack tree A is defined as a tuple $\{N, Child, Top_event, l\} \cup \{AT_elements\}$ where, N is a finite set of nodes in the attack tree; $Child: N \rightarrow N^*$ maps each set of nodes to its child nodes; Top_event is an

¹To address the disruption in user safety, we consider cybersickness as the root node of the attack tree and thus it is named as safety attack tree from here on.

unique goal node (mostly the root node) in the attack tree for the attacker where $Top_event \in N$; l : is a set of labels for each node $n \in N$; and $AT_elements$: is a set of elements in an attack tree A .

Attack tree elements: Attack tree elements aid in generating an attack tree and are defined as a set of $\{G \cup L \cup BAS\}$ where, G represents gates; L represents leaf nodes, BAS represents basic attack steps. Following are the descriptions of each of the AT elements. The *leaf nodes*, the *gates* and the *BAS* connected in the attack trees are termed as *attack tree elements*.

Basic attack step (BAS): BAS represents the *leaf nodes* of an attack tree [22]. To explore dependencies in attack surfaces, attack trees enable sharing of subtrees. Hence, attack trees are often considered as directed acyclic graphs, rather than trees [17].




Attack tree gates: Given an attack tree A , we formally define the attack tree gates G where,

$$G = \{OR, AND, SAND\}^2. \quad (2)$$

With the capability of a multi-level threat scenario in attack trees, each level in the attack tree can be modeled using gates

²We limit our attack tree to the modeling of these three gates, however attack trees can adopt any other gates from the standard fault trees.

AND, OR and SAND (Sequential AND) gates.

-  **AND:-** An AND gate is disrupted when all its child nodes are disrupted,
-  **OR:-** An OR gate is disrupted if either of its child nodes are disrupted.
-  **SAND:-** A Sequential AND (SAND) gate is disrupted only when all its child nodes are disrupted from left to right. To elucidate, the gate is disrupted using the condition: the success of previous step determines the success of the upcoming child node.

Using these gates, we define the *intermediate nodes* of an attack tree. The output nodes of the gates G in an attack tree A are defined as *Intermediate nodes* (I), which will be located at a level that is greater than the leaf nodes.

Attack tree leaves: Given an attack tree A , we formally define the *attack tree leaves* $L_{node} = \{BAS \cup \text{simple leaf nodes}\}$. In other words, L_{node} is the terminal node with no other child node(s) which is either modeled as BAS or a simple leaf node (modeled with exponential distribution) of the attack tree. For an attack tree A , we assumed the attack duration to have an exponential rate such as

$$P(t) = 1 - e^{-\lambda t} \quad (3)$$

where, λ is the rate of exponential distribution. We use the exponential distribution because of its tractability and ease of handling, since they are defined by a single parameter.

B. Threat modelling using attack trees

As part of our solution, we generate an attack tree to find the potential threats (security, privacy) that induces cybersickness (user safety) in the VR application case study (from 1) as shown in Figure 3. The safety attack tree contains root node as cybersickness which branched out to intermediate nodes such as flicker, timelag, disruption in tasks, simulator sickness. The intermediate nodes serve the purpose of establishing the relationship of root node to leaf nodes which are basically the basic attack steps to disrupt cybersickness. Some of the intermediate nodes shown in Figure 3 are *low bandwidth*, *transfer delay* and so on. We continue the branching of intermediate nodes until no further division is not possible, i.e., which terminates as leaf nodes. An example how security, privacy threats can cause cybersickness apart from physiological conditions is shown in Figure 3. To elucidate, *DoS* being a security threat and *disclosure of information* a privacy threat can disrupt $\{Timelag\}$ and $\{taskdisruption\}$ respectively, thus leading to cybersickness (CS). The nodes in the generated safety attack tree are abbreviated due to space constraints. However, a detailed description of the leaf nodes of the AT are given in Table I along with the abbreviations.

In addition to this, we also describe the abbreviations of the root and intermediate nodes of the safety AT in Table II. Using this generated safety attack tree we discuss the process of converting an AT into STA to perform stochastic model checking in the next section.

C. Translation of attack tree into STAs

In this section, we discuss about translating the leaf nodes, and the intermediate nodes which are the output events from the attack tree gates into individual STAs. An overview of the translation approach is shown in Figure 4. As mentioned earlier, these obtained STAs are used for performing model checking to verify the cybersickness metrics formalized as SMC queries. The generated STAs communicate using proper broadcast signals namely $\{initiate, fail\}$. The child nodes are activated when an *initiate* signal is broadcasted from parent node. On the other hand, *fail* signal indicates disruption of a node, which is usually broadcasted to the parent node. To illustrate the translation approach on the safety attack tree, the equivalent STA for the root node (cybersickness) is shown in Figure 5. STA broadcasts *initiate* signal and waits for the *fail* signal from the child nodes to disrupt cybersickness node. In addition, we declare $x = 0$ as a clock to keep track of the global time.

Secondly, the STA for intermediate node *Timelag* shown in Figure 3 with AND gate, which waits for the disrupt (*fail*) signal of both *Low bandwidth* ($A4$), *Transfer delay* ($B1$) is shown in Figure 6. The *Timelag* node gets disrupted once the fail signal is received from its child nodes. Now, the child node *Transfer Delay* ($B1$) with an OR gate is disrupted, when any of its child nodes *Features not working* ($B2$), *Response Time* ($B3$) sends a *fail* signal as shown in figure 7. Similarly, the node *Features not working* ($B2$) with a SAND gate is disrupted only if its child nodes *Unauthorized access* ($B4$), *Data tampering* ($B5$) send the *fail* signal in a sequential manner as shown in Figure 8.

STAs for the remaining nodes in the VRLE's safety attack tree are generated including the simple leaf nodes and BAS as mentioned earlier. Thus, we construct equivalent STA for the safety attack tree by composing iteratively into a network of STA known as NSTA [21]. We model these BAS to showcase the logical steps taken by an attacker to disrupt its parent node using attributes such as probability distribution value and rate of exponential.

Attacker Profile: To model the BAS of the safety attack tree, we utilize the attacker profiles as discussed in [23], [22]. We use such attacker profiles which gives a quantitative evaluation about several attributes such as: time, cost, skill and resources required to perform an attack. Based on the steps taken by the attacker of a skill level, the number of resources required, time taken to execute an attack and cost incurred for the attacker varies [24] [25] as shown in Table III. Attacks such as DoS, Impersonation, SQL injection, control of users are profiled as shown in the Table III. Any attack tree evaluation that incorporates attacker profiles will allow

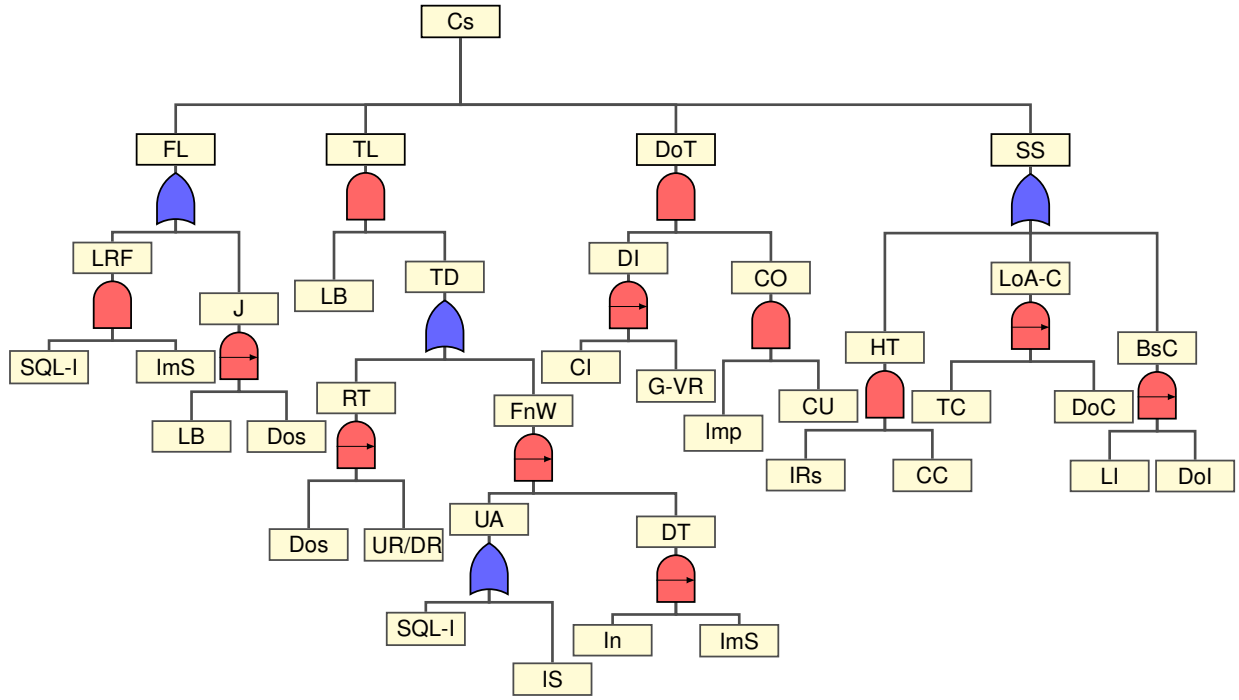


Figure 3: Safety attack tree for VR application case study: vSocial

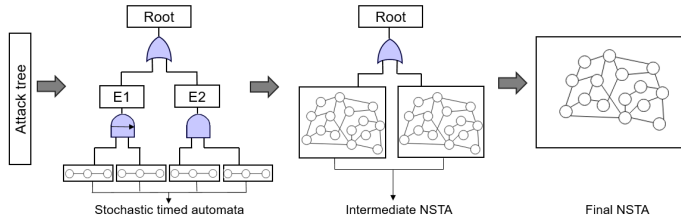


Figure 4: Framework for translation of attack trees into network of stochastic timed automata.

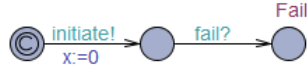


Figure 5: STA for root node (cybersickness) in attack tree.

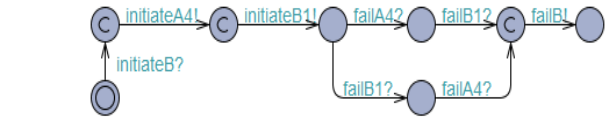


Figure 6: Stochastic timed automata for AND gate in attack tree.

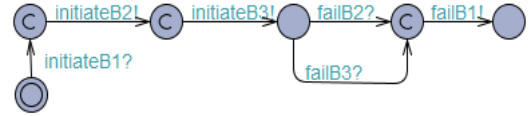


Figure 7: Stochastic timed automata for OR gate in attack tree.

us to get an indepth analysis regarding the most vulnerable components of the AT considered.

Modeling of basic attack step (BAS): BAS are the malicious atomic steps taken by the attacker to successfully disrupt a leaf node using an attacker profile. BAS are equipped with exponential distribution, discrete probabilities [17]. In addition, probability weights and rate of exponential are distributed over the edges and nodes respectively, to evaluate disruption values over various paths taken by attacker. For every attack scenario in BAS, the steps taken by the attacker are different depending on the attacker profile. In our study, we consider BAS for following leaf nodes: *Impersonation*, *SQL injection*, *Insert malicious scripts* and *Denial of service*. For instance, Figures 9 and Figure 10 discuss about the BAS for the leaf nodes impersonation, SQL

injection respectively. STA for impersonation gets activated after receiving *initiate* signal from the parent node. STA are equipped with probability weights (w_1 , w_2), rate of exponential represented as λ . In impersonation, after getting elevated access to the network, attacker chooses a path depending on the skill and resources available. Here, rate of exponential values at the intermediate nodes affects the disruption of top node in attack tree accordingly. After getting elevated access to the network, attacker can perform spoofing or session hijacking attacks with probability $w_1/w_1 + w_2$, $w_2/w_1 + w_2$ respectively. After exploiting one of the paths given in the STA, *fail* signal is sent to the parent node representing disruption of the impersonation node in attack tree.

Similar to the BAS explained for impersonation, we describe steps taken by the attacker for SQL injection in Fig-

Table I: Description of leaf nodes in safety AT

Leaf node components	Description of leaf nodes	Leaf node components	Description of leaf nodes
SQL Injection (SQL-I)	Attacker injects malicious commands in user input query using GET and POST to disclose information such as user credentials	Impersonation (Imp)	Attacker pretends to be a valid user and tries to influence user experience in VR environment.
Identity Spoofing (IS)	Attacker pretends to be a legitimate user and tries to disrupt the VR experience.	Control of user (CU)	After impersonating to VR space attacker can control the user activity.
Intrusion (In)	Attacker gets unauthorized access to user VR location.	IR sensors (IRs)	Attacker tampers the IR sensors in VR headset to disrupt the visual in VR.
Insert Malicious Scripts (ImS)	Attacker successfully adds malicious scripts in the VR environment to compromise visual or change the contents.	Change coordinates (CC)	Attacker tries to disrupt the cybersickness by changing the coordinates.
Denial of service (Dos)	Loss of availability in of user features in environment to compromise the visual or change the contents.	Track coordinates (TC)	Attacker determines the coordinates of the controllers.
Upload rate/download rate (UR/DR)	Intermittent discrepancies in network can reduce the overall bandwidth resulting in bad VR experience.	DoS of controller (Dos)	Attacker tracks the coordinates to disrupt the of the controllers.
Collect information (CI)	Information such as ip or any other confidential information of the user can be disclosed to disrupt user privacy.	Locate information (LI)	Locate the base station information which includes the information about controllers, sensors and headset information.
Goto VR space (G-VR)	Go to the user's VR space location.	Disclosure of information (DI)	Attacker discloses the component's information such as ip to disrupt the cybersickness of the controllers or change the contents.

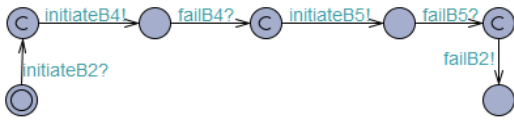


Figure 8: Stochastic timed automata for SAND gate in attack tree.

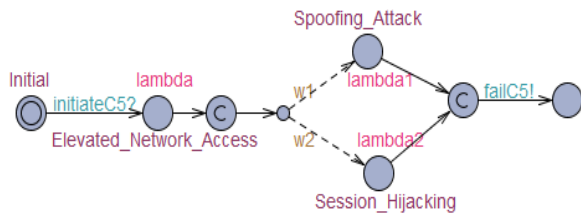


Figure 9: BAS for impersonation

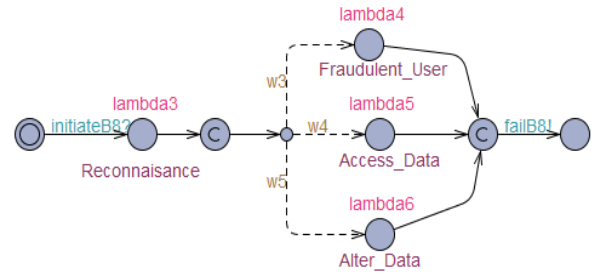


Figure 10: BAS for SQL injection

ure 10. Here, reconnaissance can be performed to find the vulnerabilities based on the value of λ given at the intermediate node. Thereafter, attacker can perform SQL injection in following ways: i) login as a fraudulent user ii) access data iii) alter data, without having authorized access. Probability

of taking different paths is described based on the probability weights i.e., $w3$, $w4$ and $w5$ distributed over the edges. After disruption, fail signal is sent to the parent node representing disruption of the respective event.

D. Safety attack tree evaluation method

In this section, we perform the quantitative analysis of several threats which are modeled as STAs. We quantitatively assess how the potential threats (security, privacy) affect the user safety i.e., causing cybersickness as shown in the safety AT. To evaluate our proposed methodology, we provide rate

Table II: Description of the Intermediate nodes for safety AT

Node	Description	Node	Description	Node	Description
Cs	Cybersickness	UA	Unauthorized Access	IRs	Infrared sensors
FL	Flicker	DT	Data Tampering	CC	Change coordinates
LRF	Low Refresh Rate	IS	Identity Spoofing	TC	Track coordinates
SQL-I	SQL Injection	In	Intrusion	DoC	DoS of controller
ImS	Insertion of malicious scripts	DoT	Disruption of userin task	LI	Locate information
J	Jitter	DI	Disclosure of information	DoI	Disclosure of data
LB	Low Bandwidth	CO	Control User	CU	Control of user
DoS	Denial of Service	CI	Collect Information	HT	Tampering of headset
TL	Timelag	G-VR	GoTo VR Space	LoA-C	Loss of availability of controller
LB	Low Bandwidth	Imp	Impersonation	BsC	Compromise base station
TD	Transfer Delay	FnW	Features not Working	-	-
RT	Response Time	UR/DR	Upload Rate/Download Rate	-	-

Table III: Attacker profiles for modeling different BAS

Type of Attack	Attacker	Skills	Resources
DoS Attack	Attacker1	High	Low (ping sweeping),
	Attacker2	Low	High (sync flood)
Impersonation	Attacker1	High	Low (spoofing attack),
	Attacker2	Low	High (Session Hijack)
SQL injection	Attacker1	High	High (Fraud. access),
	Attacker2	Low	Medium (Alter Data)
Insert malicious scripts	Attacker1	High	Low (Add URL),
	Attacker2	Low	Medium (spooky vis.)

of exponential (λ) to the leaf nodes as inputs as follows: i) for the four leaf nodes shown in Table V modeled as BAS, λ is calculated using the weights of probability of distributions for each BAS, ii) for the simple leaf nodes, we provide different values of λ as shown in Table IV representing attack duration. These λ inputs at the leaf nodes can be used to determine the probability of disruption for all the intermediate and the root nodes as mentioned. Although, such parametric values are provided for the evaluation purpose, the STA models can be evaluated using any other user specified values for different application settings.

Using the constructed STAs for the safety attack tree, we perform quantitative analysis to identify the most vulnerable components (application issues, VR components) that can act as inducing factors for cybersickness in VR application case study. This proposed methodology can be adapted to study other user safety factors in VR, however in this paper

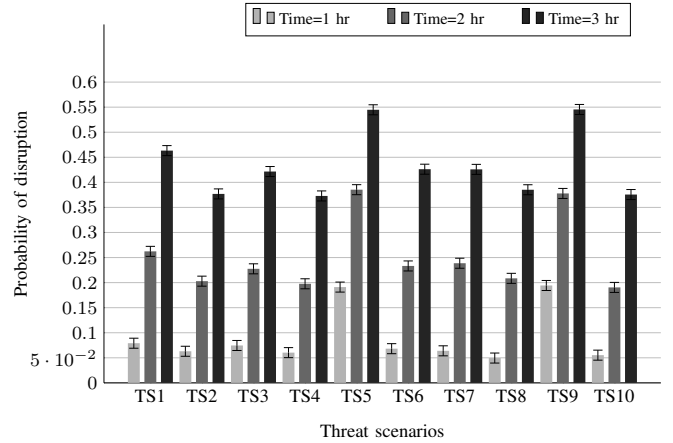


Figure 11: Probability of disruption for threat scenarios in safety attack tree with respect to time = 1hr, 2hr and 3hr.

we limit our research to cybersickness. As defined earlier, UPAAL SMC queries are used to analyze the generated STAs. While using UPPAAL to calculate the results, we use a confidence interval of 95% with an error bound ϵ of 0.01. Thus, we quantify the security and privacy threats affecting cybersickness (root node) in safety attack trees by performing analysis on: (i) individual, and (ii) combination of leaf nodes. We also study the behavior of different attacker profiles on the cybersickness. With the identified security and privacy threat factors on cybersickness, we perform a risk assessment to showcase the critical threat and components in the VR

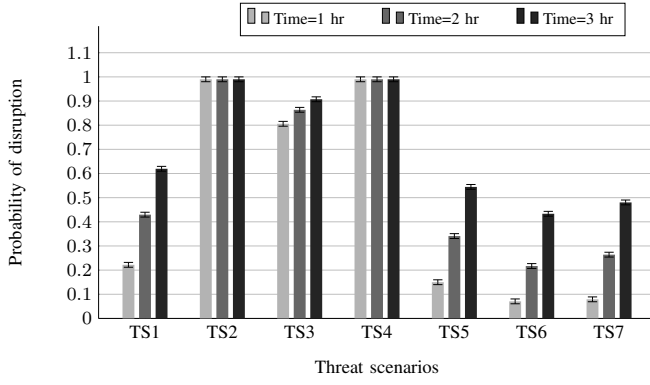


Figure 12: Probability of disruption of root node of safety attack tree due to combination of threat scenarios

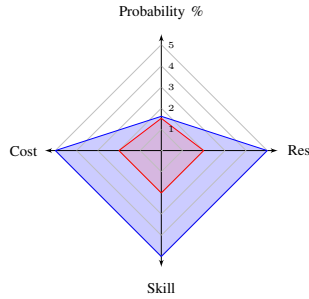


Figure 13: Probability of disruption of response time in safety attack tree for different attacker profiles.

application case study.

Analysis on individual/combination of leaf nodes: For the individual leaf node analysis on cybersickness, we utilize λ as input, while keeping the other leaf nodes at a very small positive constant (C) ≈ 0.00005 . Thus, we analyze the probability of disruption of cybersickness over different time duration. We use these provided λ values to find the individual threats that induces the cybersickness in the VRLE application.

Table IV: Values of λ given to the simple leaf nodes

Leaf nodes	
Threat scenarios	λ
Disclosure of data	0.0009298
Location information	0.0000078
DoS of controller	0.000006
Coordinates track	0.000004818
Change coordinates	0.002642
IR sensors	0.000001218
Control user	0.000006218
Goto VR space	0.0094
Collect information	0.0000071
Intrusion	0.006628
Identity spoofing	8.1219E-06
UR/DR	0.00001
Low bandwidth	0.000011219

Table V: Values of λ given to the basic attack steps

BAS	
Threat scenarios	λ
Impersonation	0.00004925, 0.00005466
SQL injection	0.0000502, 0.0000854, 0.0000492
Insert malicious scripts	0.0005389, 0.0006899
Denial of service	0.0003638, 0.0003084

On the other hand, the combination of leaf nodes that belongs to same or different sub-trees in the safety attack tree are analyzed. This analysis will aid us in determining the interplay of security/privacy/both breaches inducing cybersickness. Thus, we determine the threat vectors affecting cybersickness at a maximum level.

Study of attacker profiles: Since threat factors are dependent on attacker profiles, the probability of experiencing cybersickness is also related to the attacker profiles. Hence, we evaluate different attacker profiles from Table III to study their impact on cybersickness. Attacker profiles vary in terms of attributes such as *cost*, *resources*, *skill*, *time*, etc. Using these attacker profiles, we analyze the basic attack steps required to successfully disrupt the goal node using the available budget, resources, time and skill of the attacker. This goal node can be either root node of the attack tree or a specific component based on the attackers interest. Thus our analysis aims to provide information about the threats and vulnerable components specific to the attacker profiles and target.

Risk assessment: In order to identify the critical components against well known and relevant cyber-attack vectors, we adapt a widely accepted NIST based risk assessment procedure [26] [27] and analyze the dependability of the exemplar VR system. Using the obtained results from the individual and combination of leaf node analysis that induces cybersickness (presented in the next subsection), below are the steps that we follow to calculate the risk factor:

- 1) Assess the probability of occurrence ($P(O)$)
- 2) Assess the level of impact (I) in the event of occurrence
- 3) Calculate the risk (R) factor using $P(O)$ and I .

Thus, we formulate the risk factor as follows:

$$R = P(O) \times I \quad (4)$$

To perform the risk assessment, we adapt the semi-quantitative scale discussed in the NIST document [27] to guide us in categorizing the potential threats as 'High', 'Moderate', 'Low' risks (based on the I and $P(O)$ values). The $P(O)$ is calculated by performing quantitative evaluation on the generated STAs as discussed in next subsection. Moreover, the impact is calculated based on the guidelines given in the NIST document [27]. Using the $P(O)$ and I parameters, we calculate the risk factor for the threats outlined in the safety attack tree. The rational behind such a calculation is to get the most conservative estimate of the critical components in the

VR application.

IV. RESULTS AND DISCUSSION

A. Analysis on Individual leaf nodes

In this section, we analyze individual leaf nodes in the safety attack tree to study their effect on cybersickness over different time duration (1 hour, 2 hours, 3 hours). Due to design space constraint, we considered only a few leaf nodes in safety attack tree as shown in Figure 11 and term them as threat scenarios (TS). For this analysis, we term the considered leaf nodes as: *TS1*-{DoS}, *TS2* - {Up_Rate/DR_Rate}, *TS3*-{Low Bandwidth}, *TS4* - {Identity Spoofing}, *TS5* - {Change Coordinates}, *TS6*- {SQL Injection}, *TS7* - {Insert Malicious Scripts}, *TS8*- {Impersonation}, *TS9*- {IR Sensors}, *TS10*-{Intrusion}.

From the analysis shown in Figure 11, we identify that *TS5* - {Change Coordinates} and *TS9*- {IR Sensors} as the critical threats causing cybersickness with highest probability of disruption 0.54 at $t \leq 3$ hours. The next critical components are *TS1*-{DoS}, *TS7* - {Insert Malicious Scripts} with probabilities 0.47, 0.43 respectively. After finding the critical threats in the safety attack tree, in the next subsection we evaluate the effect of the combination of threats (security, privacy) on causing cybersickness.

B. Analysis on combination of leaf nodes

In this section, we assess the effect of combination of leaf nodes on inducing cybersickness. We use the similar analysis outlined in the individual leaf node analysis. Thus, we λ as input value while keeping other values ≈ 0.00005 at different time durations (1 hour, 2 hours, 3 hours). In this analysis, we consider combination of leaf nodes (security, privacy threats) either from the same or different subtrees of the safety attack tree and we term all such combinations as threat scenarios (TS) as shown in Figure 12. Thus, the threat scenarios used for this combination of leaf node analysis are: *TS1*- {DoS, UR/DR}, *TS2*- {IR Sensors, Change Coordinates}, *TS3* - {DoS, Low Bandwidth}, *TS4* - {SQL injection, Insert malicious scripts}, *TS5*- {Insert malicious scripts, Intrusion, Identity Spoofing}, *TS6* - {Insert malicious scripts, Identity Spoofing}, *TS7* - DoS, Impersonation as show in Figure 12. Thus, we identify that *TS2* - {IR Sensors + Change Coordinates} and *TS4* - {SQL injection, Insertion of malicious scripts} as the critical threat vectors, that disrupts the cybersickness with maximum probability of 1 within $time \leq 1$ hour. We also identify the next vulnerable combination *TS3* - {DoS, Low Bandwidth} that causes cybersickness with probability of 0.89.

In summary, with such analysis of individual and combination of leaf nodes, we deduce that IR sensors, SQL injection are the inducing factors for cybersickness. We use these identified components and threats for performing risk assessment to study the risk of these threat components on VR application.

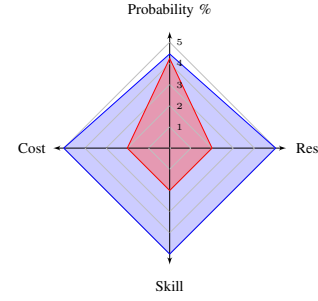


Figure 14: Probability of disruption of cybersickness for different attacker profiles.

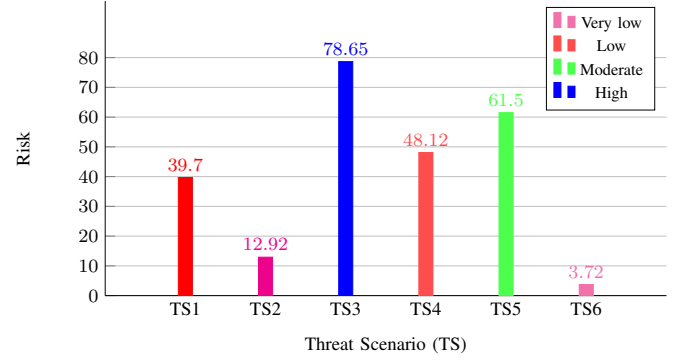


Figure 15: Risk assessment of threats affecting cybersickness

C. Analysis using different attacker profiles

In this section, we analyze how different attacker profiles and their effect on inducing cybersickness as shown in Figure 14. The attacker profiles considered for this analysis are shown in Table III. Using this analysis, we can understand the problem scope of the user safety and the significance of addressing cybersickness. From the analysis shown in Figure 14, for $time \approx 3$ hours the probability of disruption for attacker1, attacker2 is 0.51, 0.49 respectively. Similarly, we compare the probability of disruption for the specific threat/component such as {Response Time} node in the safety attack tree for different attacker profiles as shown in Figure 13. We observe this change in probability of disruption between the attackers is due to the profile attributes such as skill, cost, time, resources as shown in Table III. In the next section, we perform the risk assessment of the identified potential threat factors from the individual and combination of leaf node analysis for the given VR application case study.

D. Risk assessment

In this section, we perform the risk assessment as shown in Figure 15 using the NIST document [27] on the safety attack tree. With the semi-quantitative scale in NIST [27], [26], we calculate the risk factor as mentioned in Equation (4). Thus, we categorize on a percentage scale from 0 to 100%, where $\geq 100\%$ (very high), $> 70\%$ (high), $> 50\%$ (moderate), $> 20\%$ (low), $< 20\%$ (very low). From the column bar graph, shown in Figure 15 showcases the risk evaluation for

the identified threats in the safety attack tree. In this risk assessment, we term the resultant threat/components from the individual, combination analysis of leaf nodes as threat scenarios (TS). To compute the risk factor for such TS, we first calculate the probability of occurrence for each of these threat scenarios by applying SMC queries. We term several threat scenarios as: *TS1- {DoS+Low Bandwidth}*, *TS2 - {DoS}*, *TS3- {IR sensors}*, *TS4 - {Change coordinates}*, *TS5- {IR sensors+ Change coordinates}*, *TS6- {Insert malicious scripts + SQL injection}*. From the Figure 15, We can observe that threat scenarios *TS3 - IR sensors* has a risk value of 78.65% and thus we categorized *TS3* as high. On the other hand, threat *TS5- {IR sensor + Change coordinates}* has risk value 61.5% and thus classified into moderate risk category. In addition to this, the threat scenarios *TS1- {DoS+Low Bandwidth}* and *TS4 - Change coordinates* are classified as low risk threats, as they have risk value 39.7% and 48.12% respectively. On the other hand, the threats that fall into low risk category are: *TS2 - DoS* and *TS6 - {Insert malicious scripts, SQL injection}* with risk values 12.92% and 3.72% respectively.

Thus from our quantitative results, we observe that along with physiological conditions, the threats (security, privacy) are acting as major inducing factors for cybersickness. The identified threats/components from our quantitative evaluations are: *compromise of VR components, SQL injection and insertion of malicious scripts*. All of these three can lead to alteration of the VR contents and disruption of the orientation avatar, thereby inducing cybersickness for the considered VR application.

V. CONCLUSION AND FUTURE WORKS

Virtual reality applications are novel examples of sociotechnical systems that provides immersive user experience. With current advent of VR technologies in various domains, new applications are established without focusing on user safety as a priority. Due to overexposure in the VR system, a user may experience cybersickness which eventually disrupts the user's safety. Traditionally, it is claimed that cybersickness is caused due to physiological conditions. However, users may experience cybersickness if an attacker with malicious intent performs a cyber-attack on the VR that results into flickering of visuals, disrupting the orientation of users with spooky visuals while they move in the virtual environment. This paper proposed a methodology to identify the critical elements and potential threats that might induce cybersickness in a VR application. We utilize the attack tree formalism. For analyzing the developed attack trees based on different attacker profiles, we convert them into equivalent stochastic timed automata and apply the statistical model checking technique. Furthermore, we also perform a risk assessment on the identified vulnerable components identified using our method.

As a part of future works, we plan to cover other factors that might cause cybersickness apart from security and privacy breaches. In addition, we plan to evaluate different mitigation strategies to assess their effectiveness for mitigating cybersickness, thus ensuring a safe virtual environment for the users.

REFERENCES

- [1] A.G. Abulrub, A.N. Attridge, M.A. Williams, "Virtual reality in engineering education: the future of creative learning", *In Global engineering education conference (EDUCON)*, 2011.
- [2] S. Gregory, B. Gregory, T. Reiners, A. Fardinpour, M. Hillier, M. Lee, A. Basu, "Virtual worlds in Australian and New Zealand higher education: remembering the past, understanding the present and imagining the future" *ASCILITE*, 2013.
- [3] M. Jou, J. Wang, "Investigation of effects of virtual reality environments on learning performance of technical skills", *Computers in Human Behavior*, 2013.
- [4] Satava, Richard M, "Medical applications of virtual reality", *Journal of Medical Systems*, 1995.
- [5] Rizzo, Albert A and Schultheis, Maria and Kerns, Kimberly A and Mateer, Catherine, "Analysis of assets for virtual reality applications in neuropsychology", *Neuropsychological rehabilitation*, 2004.
- [6] Y. Cheng, S.H. Wang, "Applying a 3D virtual learning environment to facilitate student's application ability-the case of marketing", *Computers in Human Behavior*, 2011.
- [7] L. Rebenitsch, C. B. Owen, "Review on cybersickness in applications and visual displays", *Virtual Reality*, 2016.
- [8] Paja, E., Dalpiaz, F., and Giorgini, P. 2015. "Modelling and Reasoning about Security Requirements in Socio-Technical Systems," *Data & Knowledge Engineering* (98:2015), pp. 123-143.
- [9] N. Glaser, M. Schmidt, "Usage Considerations of 3D Collaborative Virtual Learning Environments to Promote Development and Transfer of Knowledge and Skills for Individuals with Autism", *Technology, Knowledge and Learning*, 2018.
- [10] K. Stanney, R. Murant, R. Kennedy, "Human Factor Issues in Virtual Environments: A Review of the Literature", *Presence: Teleoperators and Virtual Environments*, Vol. 7, No. 4, pp. 327-351, 1998.
- [11] M. Dennison, A. Wisti, M. D'Zmura, "Use of Physiological Signals to Predict Cybersickness", *Displays*, Vol. 44, pp. 42-52, 2016.
- [12] S. Martirosov, P. Kopecek, "Cybersickness In Virtual Reality - Literature Review", *28th DAAAM International Symposium on Intelligent Manufacturing and Automation*, 2017.
- [13] Krombholz, K., Hobel, H., Huber, M., and Weippl, E. 2015. "Advanced Social Engineering Attacks," *Journal of Information Security and Applications* (22:2015), pp. 113-122.
- [14] M. Mujinga, M. M. Elof, J. H. Kroeze, "A Socio-Technical Approach to Information Security", *Twenty-third Americas Conference on Information Systems, Boston*, 2017.
- [15] "Socio-technical systems: From design methods to systems engineering", *Interacting with Computers*, , Volume 23, Issue 1, Pages 4-17, 2011.
- [16] C. Zizza, A. Starr, D. Hudson, S. S. Nuguri, P. Callyam and Z. He, "Towards a Social Virtual Reality Learning Environment in High Fidelity", *15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018.
- [17] R. Kumar, M. Stoelinga, "Quantitative Security and Safety Analysis with Attack-Fault Trees", *IEEE 18th Int. Symposium on HASE*, 2017.
- [18] A. David, K. G. Larsen, A. Legay, M. Mikućionis, and D. B. Poulsen, "Uppaal SMC Tutorial", *Int. J. on Software Tools for Tech. Transfer*, 2015.
- [19] P. Bulychev, A. David, K. G. Larsen, A. Legay, D. Li, G. B. Poulsen, A. Stainer, "Monitor-Based Statistical Model Checking for Weighted Metric Temporal Logic", 2012, pp. 168-182.
- [20] P. Bulychev, A. David, K.G. Larsen, A. Legay, G. Li, D. B. Poulsen, "Rewrite-based Statistical Model Checking of wmtl", *Int. Conference on Runtime Verification*, 2012.
- [21] P. Ballarín, N. Bertrand, A. Horvath, M. Paolieri, and E. Vicario, "Transient Analysis of Networks of Stochastic Timed Automata Using Stochastic State Classes", *Proceeding of the 10th international conference on Quantitative Evaluation of Systems*, 2013.
- [22] G. Norman, D. Parker, J. Sproston, "Model checking for probabilistic timed automata", *Formal Methods in System Design*, Vol. 17, pp. 164-190, 2013.
- [23] M. Rocchetto and N.O. Tippenhauer, "On Attacker Models and Profiles for Cyber-Physical Systems", *ESORICS*, 2016.
- [24] "Attacker Classification to Aid Targeting Critical Systems for Threat Modelling and Security Review", <http://www.rockyh.net/papers/AttackerClassification.pdf>.

- [25] "D 4.4 Profiles of Cyber-Criminals and CyberAttackers", https://www.cyberroad-project.eu/m/filer_public/2016/05/02/d44_profiles_of_cyber_criminals_and_cyber_attackers.pdf
- [26] M. Dickinson, S. Debroy, P. Calyam, S. Valluripally, Y. Zhang, R. B. Antequera, T. Joshi, T. White, D. Xu "Multi-cloud Performance and Security Driven Federated Workflow Management", *Transactions on Cloud Computing*, 2018.
- [27] R. S. Ross, "Guide for Conducting Risk Assessments", *NIST SP800-30-Rev1 Technical Report*, 2012.