Zero Trust security in cloud-based simulation

Patric Stout, Tom van den Berg, Luca Morgese Zangrandi
TNO Defence Research
The Hague, The Netherlands
patric.stout@tno.nl, tom.vandenberg@tno.nl, luca.morgese@tno.nl

ABSTRACT

Information Technology (IT) security approaches traditionally attempt to translate perimeter-based security from locked doors, badges, and guns to firewalls and digital access control policies. But this only works on localized IT systems. The state-of-the-art demonstrated how unsuited this approach becomes to security in federated, multi-party, cloud-based simulation environments. Neglecting security controls within such an infrastructure may leave open chances of misbehavior and "honest but curious" behavior. Without internal security controls, parties in the simulation may be able to gather far more details than they should. For instance, a simulation component may subscribe to more simulation data than required for a correct and valid interoperation with other components. In other cases, a component may attempt to instigate information disclosing responses from other components by publishing more simulation data than necessary. In other words, a traditional perimeter-based security approach to a federated cloud-based simulation environment may allow any component to easily exfiltrate, falsify, and/or disrupt information.

In recent years the Zero Trust approach to cybersecurity has gained increasing momentum, pushing the philosophy of "never trust, always verify", and "assume breach". In essence, Zero Trust mandates that proof of trustworthiness cannot be derived from simply having access to an environment: it must be possible to verify to the most risk-relevant extent feasible that processes and entities can be trusted continuously and according to a dynamic context.

This paper explores the application of Zero Trust approaches in the context of security in cloud-based simulation. We describe a framework to tailor Zero Trust concepts to the design and implementation of security controls in an HLA based simulation environment and present the results of a field-test of these controls at CWIX 2024 in the context of a larger effort for the NATO Federated Mission Networking.

ABOUT THE AUTHORS

Patric Stout is a senior engineer in the Modelling, Simulation and Training department at TNO, The Netherlands. He has 10 years of experience in the cybersecurity domain, in, among others, detection & response, cloud-based infrastructure, and Zero Trust for SaaS products. He is pushing for making distributed simulations secure in a way that is easy to adopt and requires little technical know-how of the cybersecurity domain itself.

Tom van den Berg is a senior scientist and systems engineer in the Modelling, Simulation and Training department at TNO Applied Physics Laboratory, The Netherlands. He has over 25 years of experience in distributed operating systems, database systems, and simulation systems. His research area includes simulation systems engineering, systems of simulation systems, cloud-based and LVC simulation, and distributed simulation architectures. He is a member of several SISO Product Development Groups and participates in various M&S standardization activities.

Luca Morgese Zangrandi is a cybersecurity scientist at TNO, The Netherlands, and OASIS technical contributor to cybersecurity interoperability standards. He is involved in several research and development projects regarding, among others, Zero Trust architectures, and automation of security operations.

Zero Trust security in cloud-based simulation

Patric Stout, Tom van den Berg, Luca Morgese Zangrandi
TNO Defence Research
The Hague, The Netherlands
patric.stout@tno.nl, tom.vandenberg@tno.nl, luca.morgese@tno.nl

INTRODUCTION

Throughout the last decades, information technology architectures have progressively evolved from monolithic and static systems, toward highly dynamic and flexible environments, such as cloud-computing platforms. In these environments, an application is developed as micro-services: function-specific logic modules designed to be deployed in a highly flexible and distributed way. These flexible infrastructures drastically improve organizational IT management. There is no longer any need to purchase, scale, configure, and connect bare-metal function-specific computing machines: resources and capabilities can be generated and automatically orchestrated in a virtually limitless virtual environment, such as a cloud platform. This trend is present also in the networking domain. Approaches such as Software Defined Networking, virtual network capabilities, and data flows. These allow to manage and configure networks at machine-speed, without the need to deploy and configure specialized hardware devices.

In this context, where in the past it was possible to clearly identify organizational boundaries in an IT infrastructure, this is not the case anymore. The requirements and methodologies of securing these IT environments thus also evolve. It is no longer possible to fix and harden a cyber-physical security perimeter around all organizational IT capabilities, in “castle-walls” fashion; cyber threats can infiltrate from many directions, such as software supply chain, third party services, security-loose network segments, etc. Zero Trust emerged as an overarching cybersecurity paradigm to respond to these challenges. At the core of Zero Trust is the assumption that an attacker is already present in your (IT) environment. With this consideration, security perimeters must in principle be raised around all components within the IT environment, and the integrity and validity of interactions between components should always be verified. If this is correctly implemented, attackers will have a hard time penetrating IT components. Even if attackers manage to take over IT components, they will have a hard time in propagating their control to other components. Overall, Zero Trust architectures improve cybersecurity posture, and reduce possible impacts of IT threats.

The move towards virtualized and distributed IT environments also concerns the simulation domain. Where traditionally simulation components are deployed and federated in a “castle walled” (localized) IT environment, they are now often virtualized and transitioned to a cloud-computing platform on which they can be deployed on demand, participating as a service in some federated simulation. Cloud-based simulation components may for instance offer services to other, non-cloud-based simulation components.

If no security controls are present, parties (e.g., non-cloud-based “client” components) that join a federated simulation may be able to gather far more data than they should. For instance, a component that subscribes to more simulation data than required for a correct and valid interoperation with other components. Or a component that attempts to instigate information-disclosing responses from other components by publishing more simulation data than necessary, or perhaps tries to take ownership of certain data. Hence, security controls are essential to minimize both the opportunity for and the impact of such incidents. It is thus worth exploring the application of Zero Trust approaches and subsequent security controls in federated and cloud-based simulation.

In this paper we present Zero Trust approaches in simulation. First, we start with the larger context and main driver behind this exploration, namely the NATO Federated Mission Networking (FMN) and the Coalition Warrior Interoperability Exercise (CWIX). Next, we introduce the framework called “Zero Trust for Simulation Architectures” (ZeTSA). This framework helps in the understanding of Zero Trust notions, and how they apply to a simulation environment. Furthermore, we show the application of this framework to an HLA-based simulation environment, with the introduction of security controls in CWIX 2024. At the end of this paper, we discuss the results of CWIX 2024.

CONTEXT

NATO Federated Mission Networking

The NATO Federated Mission Networking (FMN) is a major initiative to help ensure the interoperability and operational effectiveness of the NATO coalition. The goal is "day-zero interoperability", that is, systems can safely and reliably exchange data and information supporting coalition operations right from the start – day zero – of a mission. The initiative is led by NATO's Allied Command Transformation (ACT) co-operating with NATO's Allied Command Operations (ACO). Over 35 NATO and non-NATO nations participate in FMN (NATO FMN, 2024). Besides interoperability, security is an integral part of FMN.

Development of M&S Specifications for FMN

The NATO Research Task Group (RTG) MSG-201 "Modelling and Simulation in Federated Mission Networking (FMN)" (NATO MSG-201, 2022) participates in the FMN organization as an "FMN syndicate" to support the development of M&S specifications for FMN. RTG MSG-201 has been active since 2022 and has developed FMN Procedural Instructions for Mission Rehearsal and FMN Service Instructions for M&S. Proposed updates include M&S for Collective Training. All planned to be incorporated in future FMN spirals. The currently developed FMN M&S specifications reference several key NATO M&S standards for use in FMN, such as the High Level Architecture (HLA) (NATO STANAG 4603, 2015), and the NATO Reference Architecture for Distributed Synthetic Training (NATO DST, 2022). The RTG MSG-201 itself acts under the umbrella of the NATO Modelling and Simulation Group (NATO STO, 2024), with members from various NATO nations.

Coalition Warrior Interoperability Exercise

The Coalition Warrior Interoperability Exercise (CWIX) (NATO ACT, 2024) is an annual and major NATO interoperability event and provides an environment where FMN interoperability specifications can be tested against experimental and near-fielded capabilities. CWIX fosters innovation by identifying and solving interoperability shortfalls, experimenting with alternative approaches, and exploring emerging technologies.

MSG-201 uses CWIX as a venue to explore requirements for inclusion in the FMN M&S specifications and to conduct interoperability tests between federated simulation systems, and between C2 and simulation systems. CWIX provides MSG-201 a structured test approach in combination with a test environment that enables the evaluation of system interoperability.

One of the MSG-201 objectives for CWIX 2024 is the exploration of security interoperability requirements in support of Mission Rehearsal and Collective Training with M&S. This exploration fits in the context of a wider FMN objective, namely, to introduce a Zero Trust Architecture for FMN. The MSG-201 objective and CWIX provide an excellent setting to explore the application of Zero Trust approaches and security controls as described in this paper. The outcomes can also be used to improve the M&S specifications with security related requirements.

High Level Architecture

The High Level Architecture (HLA) (IEEE 1516, 2010) is an international standard for the development of a distributed simulation environment and is a key standard in the FMN M&S specifications. The HLA is focused on interoperability between various types of simulations and promotes reuse of simulations and their components. The HLA follows two general design principles:

- *modularity*: simulation components (federates) are composed into larger systems (federations) to obtain a specific functional behavior.
- *separation of concerns*: the functional behavior of the components (federates) is separated from the supporting communication infrastructure via a well-defined interface.

In the terminology of the HLA, an individual simulation is known as a "federate". A federate may be a data collector, a simulator, or a viewer. The collection of federates brought together to form a synthetic environment is known as a "federation". Figure 1 provides an example of an HLA federation, where live participants, a simulation, and support tools interact through a Run Time Infrastructure (RTI).

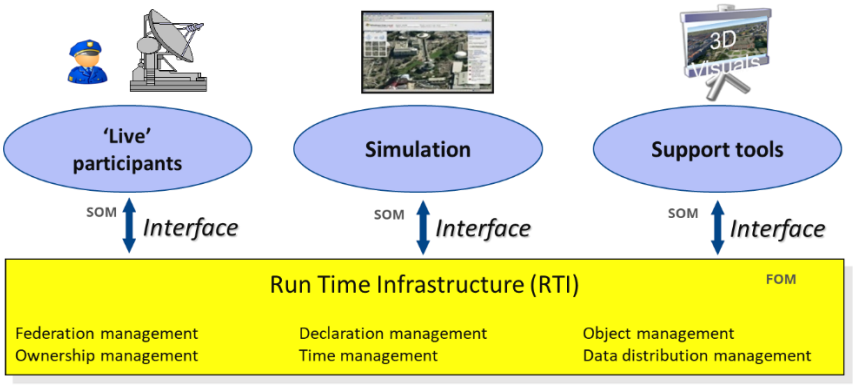

Figure 1: A graphical view of the HLA: federates operate together through a common Run Time Infrastructure (RTI).

The RTI provides several Application Programming Interface (API) service groups used by a federate to interact with the RTI. The common interpretation of a shared data model, called the Federation Object Model (FOM), allows federates to interact within a single synthetic environment. All possible data exchanged by federates in a federation is captured in a FOM. The data exchanged by an individual federate is captured in its Simulation Object Model (SOM). And RTI management data is captured in the Management Object Model (MOM). Both FOM, SOM, and MOM use the same XML schema, as defined by the HLA standard.

Over time several HLA versions have been released. The most recent version is HLA 1516-2010 (HLA Evolved). The next version is HLA 1516-202X (HLA 4), expected to be released in 2024. This version includes the specification of a Federate Protocol and an Authorizer API, both relevant for security. An implementation of the HLA 4 Federate Protocol and Authorizer API is already available for the Pitch RTI (Pitch, 2024).

ZETSA FRAMEWORK

The ZeTSA framework maps existing Zero Trust guiding principles and approaches to their application in simulation architectures. Although the framework can be applied to any simulation architecture, it is mostly meant to help securing cloud-based simulation concepts such as M&S as a Service (MSaaS) (NATO, 2017). The ZeTSA aims to provide a baseline understanding of what Zero Trust means in terms of security requirements, and then maps these requirements to Operational Objectives. The framework is not meant to be exhaustive, but rather a helpful reference point to facilitate a common ground for discussion on Zero Trust concepts and their contextualization in simulation architectures.

Building the ZeTSA Framework

The ZeTSA Zero Trust Operational Objectives are synthesized from a semantic aggregation of Zero Trust concepts provided by three authoritative Zero Trust reference sources: The Zero Trust Maturity Model, from the US Cybersecurity and Infrastructure security Agency (CISA) (Cybersecurity Division, 2023); the Special Publication 800-207 from the US National Institute of Standard and Technology (NIST): Zero Trust Architecture (Rose, Brochert, Mitchell, & Connelly, 2020); and the Zero Trust Reference Architecture from the USA Department of Defense (USA DoD, 2022). Although these reference documents are meant for more generic Zero Trust applications, they are useful to compose a framework that is designed around simulation architectures.

The CISA Zero Trust maturity model presents how to implement Zero Trust in an organization across five pillars: Identity, Device, Network and Environment, Application Workload, and Data. For each pillar, Zero Trust can be built via progressively enhancing functions of Visibility and Analytics, Automation and Orchestration, and Governance. For each function in each pillar, the maturity model lists what are their traditional, advanced, and optimal implementations towards achieving Zero Trust. In ZeTSA, we focus on what the maturity model identifies as the optional implementations for such functions.

The NIST Special Publication 800-207 presents the Zero Trust Architecture. The document expresses that an ideal Zero Trust implementation respects seven tenets, shortly summarized hereafter: (1) all data sources and computing services are resources; (2) all communications are secured; (3) access to individual resources is granted on per-session basis; (4) access control is dynamic and contextual; (5) the enterprise monitors and measures integrity and security posture of all assets; (6) authentication and authorization is dynamic and strictly enforced before any access is granted; (7) the enterprise has as-extensive-as-possible visibility over its assets and systems status. The publication proceeds to present different architecture topologies, use cases, and technologies to implement Zero Trust. In the ZeTSA framework, we list the tenets from 2 to 7, as the first tenet is assumed.

The USA DoD presents a Zero Trust Reference Architecture. The document lists tenets (foundational elements) and pillars (focus areas) for Zero Trust implementation. It provides a more explicit high-level guidance via presenting Zero Trust taxonomies of capabilities, which show relationships of technical functions in four different areas (capabilities): (1) authentication and authorization; (2) workload and data; (3) analytics and orchestration; (4) trust enabling capabilities. In the ZeTSA framework we refer to each taxonomy with its respective functions.

The ZeTSA Framework

By observing the notions of Zero Trust from the three documents, it is possible to identify general areas that all address, and to which each proposed function, tenet, and capability, can be assigned. We thus derive a core of five Operational Objectives (OOs) to achieve Zero Trust, which build up the ZeTSA framework:

1. *Coordinated orchestration and governance*, concerning security policy definition and administration.
2. *Observability, monitoring and telemetry* of an IT infrastructure in its assets, and how the assets interact.
3. *Command and Control* to orchestrate security functions.
4. *Dynamic, least-privileges, and specification-based Authentication, Authorization, and Accounting* (AAA): meaning the ability to verify asset identity (authentication), validate their operations (authorization), and register all their interactions (accounting) - “never trust”.
5. *Trust and systems verification*: meaning, continuously evaluate AAA rules - “always verify”.

These OOs can be mapped onto the NIST Cyber Resilience Framework (CRF) (Ross, Pillitteri, Graubart, Bodeau, & McQuaid, 2021). Table 1 shows the relation between the Zero Trust Operational Objectives in ZeTSA (column 1), and the NIST CRF Design principles (column 2). To these design principles in turn follow applicable techniques, and related cyber resiliency approaches (column 3).

Table 1: Relation of the ZeTSA OOs to the NIST CRF design principles and resiliency approaches.

ZeTSA OO	NIST CRF Design Principles	NIST CRF Approaches
1. Coordinated Orchestration and Governance	Limit the need for trust, layer defenses and partition resources, maintain situational awareness.	Adaptive management, pre-defined segmentation.
2. Observability, Monitoring and Telemetry	Maintain situational awareness, leverage health and status data.	Dynamic resources awareness, mission dependency and status visualization, monitoring and damage assessment, sensor fusion and analysis, forensic and behavioral analysis.
3. Command and Control	Change or disrupt attack surface, contain, and exclude behaviors.	Orchestration, dynamic reconfiguration, dynamic segmentation.
4. Dynamic, least-privileges, and specification-based AAA	Contain and exclude behaviors, control visibility of resources and usage, manage resources in risk-adaptive way.	Calibrated defense in depth, trust-based privilege restriction, attribute-based usage restriction, dynamic privileges.
5. Trust and Systems Verification	Continuously determine trustworthiness.	Consistency analysis, self-challenge, integrity checks, provenance tracking, behavior validation.

Starting with the ZeTSA framework, it is thus possible to investigate if and how the cyber resilience approaches implementing Zero Trust are adopted in simulation architectures. Ultimately, incorporating such approaches in a simulation environment would advance its security posture. The ZeTSA framework contains a more detailed mapping of all three documents onto these OOs, including visual images. Out of brevity these are omitted in this paper.

APPLICATION OF THE ZETSA TO THE HLA

To gather insights and knowledge on applying the ZeTSA framework to a simulation architecture, HLA is used to demonstrate what security controls can be added to increase trust of federates. In this chapter we introduce initial security controls for an HLA-based simulation environment. With these security controls, an HLA-based simulation environment becomes more resilient against cyberattacks, particularly in the context of cloud-based simulation.

Basic Zero Trust Controls

In the ZeTSA framework one can identify a few Zero Trust security controls which can be considered "the basics": authentication (who you are), authorization (what you can access), and encryption. Without these three controls, all other controls are far less meaningful, and as such, it is mandatory to have these in any good Zero Trust system. As example, one could have authentication without encryption. But in the Zero Trust concept of "assumed breach", one must assume a malicious actor is active on that same network. It would be trivial for such a party to capture the unencrypted authentication and reuse it for their own access.

When looking at the ZeTSA framework, applying these three controls alone would add security posture for all five of the Operational Objectives: limit the need for trust (OO #1), observe (OO #2), reduce attack surface (OO #3), apply least-privileges (OO #4), and trust (OO #5). In the current HLA revision (HLA Evolved) these three basic security controls are difficult to implement, as neither encryption nor authentication is part of the standard. It is up to the HLA implementation to either have it or not. With the upcoming HLA 4 this changes. HLA 4 provides the ability to add authentication and authorization logic via a standard API. Encryption of simulation data is still not mandatory in HLA 4, but this can be added on top of the "Federate Protocol" (new in HLA 4) (Moller, 2022). For example, by using Transport Layer Security (TLS), a best practice for encrypted connections between two parties.

For authentication, there are several options. The two most commonly used authentication methods are Security Assertion Markup Language (SAML) (SAML Standards, sd) and OpenID Connect (OIDC) (OpenID Specifications, sd). Both methods introduce an Identity Provider (IdP). An IdP stores and manages all user identities. A user or machine can identify itself against the IdP and receive an access token. This access token can be used as proof-of-access to systems which accept such tokens. Both SAML and OIDC go in detail how such tokens can be validated and trusted by other parties. Embedded in this token, whether it be a SAML or OIDC token, can also be information for access control. For example, "which federation the federate has access to", "which RTI operations can be performed" (e.g., are you allowed to create a new federation?), etc.

Within the context of HLA this means that the HLA RTI does not have to do any user management; it only needs to know which tokens to accept. By centralizing the IdP, strong controls can be put in place around the IdP to ensure it can be trusted and its information is kept secure. For example, it could be placed in a trusted, more closely monitored, network, with only pin-hole access for authentication and to retrieve access tokens. This alleviates the pressure from components such as the RTI, as keeping user information secure is not an easy feat.

Figure 2 shows the typical pattern in establishing a trusted connection with the HLA RTI using an IdP with the OIDC Client Credentials Flow:

1. Every new connection is set up to use encryption.
2. The HLA Federate connects with the IdP to prove its identity and retrieves an access token upon success.
3. The HLA Federate can exchange this token with the HLA RTI, which the RTI on its turn can validate with the IdP for validity.
4. As the RTI now knows the token is valid, and as such, the federate is authenticated (with the IdP), can finish setting up the connection, and trust the HLA federate to be a valid entity.

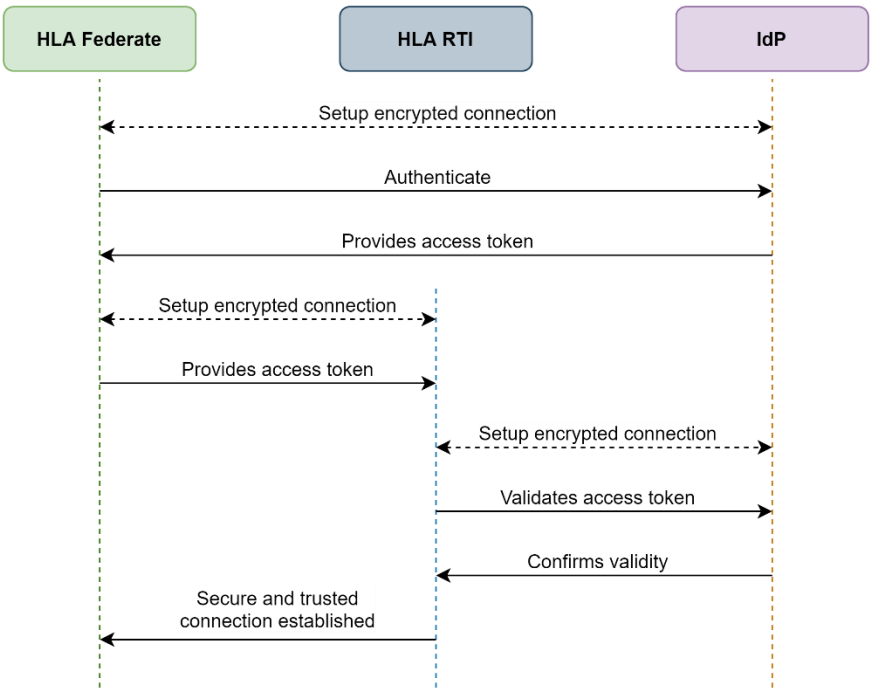

Figure 2: Common flow to setup a secure connection between Federate and RTI.

The HLA 4 Federate Protocol, extended with encryption, authentication, and authorization controls, makes the communication between federate and RTI, even over a potentially unsecure network, more secure.

Validation of Federates

These controls concern the security validation of federates. On one hand by applying the objective of "least-privileges", on the other hand by monitoring more closely what a federate is doing and alerting on suspicious behavior. With the basic security controls, a federate has a connection to the RTI that can be trusted. Next step is to ensure the federate itself behaves within the defined constraints. For example, a "supply-chain-attack" could cause a federate to show unexpected behavior, where it could be used to exfiltrate information from a simulation to a malicious actor. Or maybe the federate just has a programming error, causing disruptions in the simulation.

HLA 4 already makes a step towards adding authorization controls in the form of an HLA Authorizer. The HLA Authorizer validates if a federate is allowed to connect to the RTI and is allowed to create or join a federation execution. However, the HLA Authorizer does not prevent a federate from sending or receiving certain (undeclared) types of messages (e.g., to solicit for responses) or prevent a federate from performing certain federation management functions (e.g., to eject another federate from the federation execution).

To move one step further, we suggest additional validation to determine if a federate is allowed to send or receive certain types of messages. This can for example be implemented by using a "validated" HLA SOM of each federate. A federate must already provide its SOM as per HLA rules, so the HLA SOM of each federate should be available up front and be part of the federation agreements. Each SOM should be accessible to all members of the federation, so they too can check if another federate is performing certain un-agreed operations or requesting access to objects and/or attributes it should not have access to. For instance, sending a MOM message to eject a federate.

With this addition to the HLA Authorizer as specified in HLA 4, one can apply the least-privilege concept of Zero Trust, where a federate has just enough access to execute its function, but nothing more. This means that even if a federate is compromised, the attack surface is as small as possible. This adds another layer of defense. This again contributes to all five operational objectives of the ZeTSA framework.

User Access via Applications

A federation typically consists of a mix of federates. Besides the actual simulations, this includes, for instance, viewers, simulation management applications, data recorders, and mediation services like gateways. All these different types of federates need to apply Zero Trust concepts for the system as a whole to be more resilient against malicious actors. With the security controls discussed in the previous sections, these federates require valid credentials to access the federation execution.

User facing capabilities, such as a viewer, concern users that do not directly access the federation execution. A user interacts with the user facing part of the federate, whereas the federate interacts with the RTI, subjected to the various security controls discussed earlier. Per Zero Trust principles, such user facing capabilities also need security controls for the user interface. Otherwise, a malicious actor can use these federates to gather (possibly restricted) information about the simulation. To prevent this, role-based access controls, for instance, should authenticate users, and restrict their access to information.

It depends on the federate type how to implement this. For web applications for example, the OIDC Authorization Code Flow can be used to get a valid token. The web application should validate this token and use it to check if the user has access to the resources of the federate. The same IdP can be used to manage credentials and provide access tokens for both users and federates.

EXPERIMENTATION IN CWIX

CWIX is used as venue to explore security requirements for the use of M&S in FMN. This chapter provides an overview of the CWIX 2024 experiment and presents the results.

Cloud Platform

A suite of HLA-based simulation applications and services is provided (at CWIX) on demand from a cloud platform, available to client applications to perform for instance a Mission Rehearsal (see Figure 3). This Cloud Platform provides, amongst others, a Computer Generated Forces (CGF) service (to generate simulation entities based on an initial scenario), a simulation management application (to start and stop the simulation time, and to upload the initial scenario), a scenario initialization service (to provide the initial scenario to any consumer), and an HLA RTI service (to connect all services and applications in an HLA federation execution).

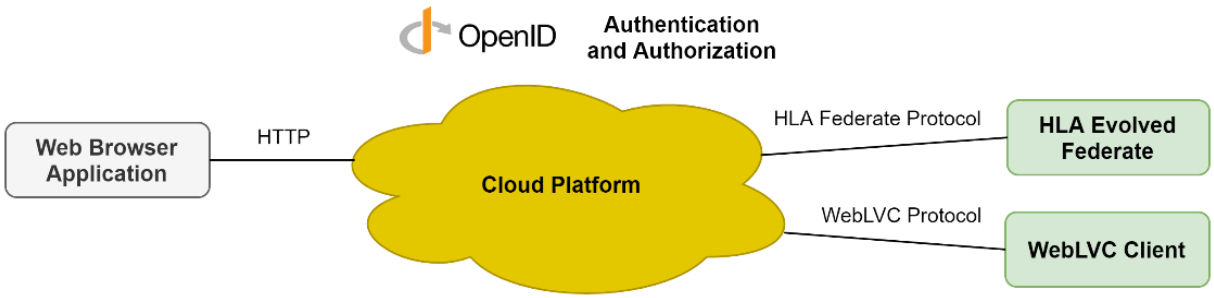

Figure 3: Cloud Platform with a suite of services offered to client applications: HLA Federates and WebLVC Clients.

Client applications can access the services provided by the Cloud Platform by connecting to the HLA RTI and joining the federation execution. Client applications are HLA Evolved Federates or WebLVC Clients, using the HLA Federate Protocol (Moller, 2022) and the WebLVC Protocol (SISO, 2022) respectively to communicate with the services. The simulation management application can be accessed from a web browser. All access to the Cloud Platform is secured with OIDC, including access from the Web Browser Application.

Experiment Questions

The experiment is about applying security controls on the access to services by client applications. The goal is to minimize the effort on the client side to maximize adoption.

There are two main questions that we try to answer with this experiment:

1. What is the effort required to adapt an HLA Evolved Federate or a WebLVC Client to the Zero Trust controls?
2. Do the Zero Trust controls work and deliver a more resilient simulation environment against cyberattacks?

Experiment Setup

To make the transition to security controls as easy as possible in this experiment, (a) no encrypted data communication is used, and (b) access tokens are provided to HLA federates in advance. WebLVC clients retrieve the access token from the IdP themselves. For the IdP, an OIDC implementation is used in CWIX 2024. The main components and interfaces relevant to the experiment are shown in Figure 4.

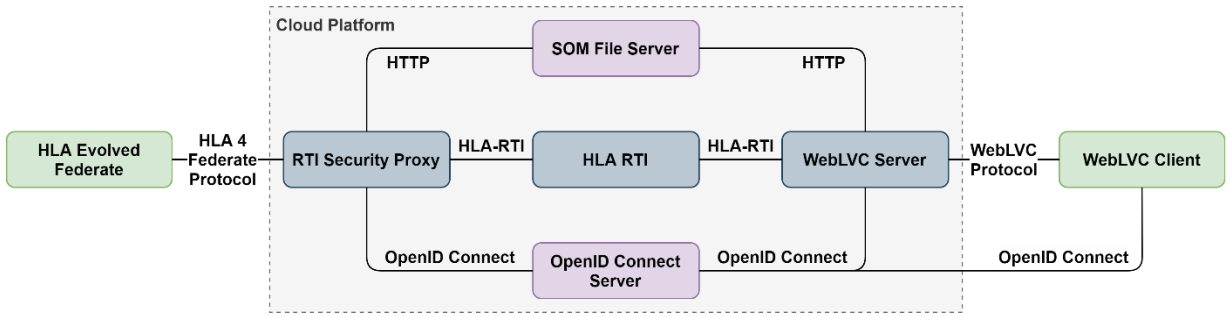

Figure 4: Experiment components and interfaces.

The HLA Evolved Federates (and their physical location) in this experiment are: MASA SWORD (JFTC/POL) (MASA, 2024), IABG KORA (JFTC/POL) (IABG, 2024), and PEO STRI OneSAF (Orlando/USA) (OneSAF, 2024). The WebLVC Clients include Symlythics Mission Command Agent (JFTC/POL) (Simlythics, 2024). The Cloud Platform with all its components is located in Maasland/NLD. The TNO RTI Security Proxy and the TNO WebLVC Server both include an Authorizer for the validation of clients. The Authorizer consults the OIDC IdP to verify access tokens and retrieves validated SOM files from the SOM File Server for interest validation. The File Server uses a secure repository. The HLA RTI is the Pitch RTI (Pitch, 2024), including a Federate Protocol Server integrated in the TNO RTI Security Proxy.

The Web Browser Application in Figure 3 (and for simplification excluded in Figure 4) is a simulation management application, providing a web-based UI from where the user can initialize the simulation, issue tasks, and view the simulation entities on a map. The user is required to provide login credentials to access the information in the application's back-end components deployed on the Cloud Platform. The back-end components are HLA federates. The security controls involve the execution of the OIDC Authorization Code Flow behind the scenes, with the IdP providing the necessary authorization code and access token. The flow ensures that the user is authenticated and authorized to perform certain operations and access the information.

The access token for HLA federates and WebLVC clients includes a set of five claims specifying: (1) the federations that the client is permitted to join, (2) the federate names that the client is permitted to use, (3) the federate types that the client is permitted to use, (4) the operations that the client is permitted to perform (e.g. create federation execution), and (5) the permitted interest in object class attributes and interaction classes (these interests refer in fact to the HLA SOM of the client). All these claims are mandatory and validated by the different Zero Trust components.

The following requirements are put in place on the clients:

- An HLA federate shall provide the access token upon connecting to the RTI.
- A WebLVC client shall retrieve the access token from the IdP and shall provide this access token in the initial connect message to the WebLVC Server.

Test Cases

The experiment involves the execution of a set of test cases in which each client connects to the RTI with a test case specific access token. The claim values in the token vary across the test cases. For the interest claim the access token provides the client either: full access (can publish/subscribe any HLA FOM class), limited access (can only publish/subscribe to certain HLA FOM classes), no access (cannot publish/subscribe any HLA FOM classes), and only own-SOM access (can only publish/subscribe classes in accordance with the previously provided own HLA SOM). Other claim values in the test cases concern the permitted federate names and federate types, and the federations allowed to join.

Experiment Results

The adoption of the Zero Trust security controls by the CWIX test partners was quick: most of them could make the required changes in their application in a day or less and could join the federation execution as they normally would. This is important, as the less friction this creates, the more likely it is these controls get adopted.

Required adaptations were minimal and include:

- For the HLA federate: adding the access token to the RTI connect invocation and configuring the application to use the Pitch HLA Federate Protocol Client LRC.
- For the WebLVC client: adding logic to request a token from the IdP and passing this token to the initial WebLVC connect message.

The results are in summary:

- HLA federates and WebLVC clients quite seamlessly transitioned over to OIDC security controls. The controls were in place for all clients throughout the CWIX exercise in all test cases performed by the test partners. This also included a larger Mission Rehearsal event test case involving all clients and lasting several hours.
- The class and attribute-based authorization using the OIDC Client Credentials Flow worked well. A client received publication and subscription authorization errors in accordance with what was permitted. Depending on the client, on an authorization error, it either continued with what was permitted or terminated entirely.
- Also, the other validations worked well. A client could only join a federation it was permitted to join. The federate type of a federate could not always be changed, but this was resolved by adding more permitted type names to the access token.
- User authentication and authorization was demonstrated successfully for the simulation management application using the OIDC Authorization Code Flow.

There were no specific test cases on performance and stability. However, generally the impression was that the initialization phase of each federate took relatively more time, and that the simulation execution phase was more in accordance with normal performance. This is most likely due to the HLA 4 Federate Protocol, the geographic distribution of the clients and the Cloud Platform, and the network topology (i.e., not related to the security controls).

DISCUSSION AND FUTURE WORK

The HLA 4 standard does not require encryption for the communication between federates and RTI using the Federate Protocol (the standard marks TLS support as optional). This means that it is up to the suppliers of RTIs to either have this or not. We hope this paper gives enough motivation to RTI suppliers to include the possibility to enable encryption for the Federate Protocol.

The HLA 4 Authorizer API is a good start to add security controls in a simulation environment. However, further security controls are needed on the exchanged message types. The proposed solution in this paper is to extend the Authorizer API with the ability to authorize publish/subscribe operations using a validated HLA SOM. Having this extension to the HLA 4 Authorizer API, in combination with the possibility to enable encryption, would mean the controls discussed in this paper can fully be implemented within the HLA 4 specifications.

Additionally, the authors of this paper propose standardizing the type of IdP to use for these kinds of security controls, namely OIDC, as well as the claims in access tokens. Although SAML is equally viable, OIDC is already widely adopted by the Internet. This would create less friction when combining HLA-based simulation environments with websites that give access to parts of the simulation environment.

The controls explored in this paper are so-called active controls: they can actively refuse connection or commands based on the information available. But such controls can make mistakes. There is also a need for more passive controls, like monitoring and logging. These give a better after-the-fact view of what happened, and what the impact would be if the active controls failed to do their job. More work is required to investigate the feasibility of this, and the added value of these controls in simulation environments.

Adding security to any system always impacts performance – it is a tradeoff; this impact must be benchmarked and evaluated. This work has yet to be done. Furthermore, the focus has been on HLA-based simulation environments, and not their connection with other systems, like C2-mediation services and C2 systems.

CONCLUSIONS

In a world where "castle wall" defenses are no longer sufficient to connect different environments (like an on-site simulation environment to a cloud-based one) together, there is need for better controls to keep those setups safe and secure. As the Internet adopted Zero Trust in a response to this threat, this paper investigated whether the same concepts can be used for simulation environments.

This paper introduced the ZeTSA framework to apply Zero Trust concepts to a simulation environment. Identifying Zero Trust Operational Objectives and their implementation techniques, the ZeTSA was applied to HLA-based simulation environment to improve its security posture. To ensure the viability of this solution, it has been field-tested during CWIX 2024.

The field-test was a great success and shows that Zero Trust can be introduced in an HLA-based simulation environment with little to no friction. In return, cloud-based simulation environments can connect to an on-site simulation environment in a safe and secure manner, where federates on both sides can be trusted to be authenticated and authorized and abide by agreed-upon behavior. The tests at CWIX provided useful insights into potential security related requirements for the FMN M&S specifications. Requirements for HLA federates and WebLVC clients include: obtain an access token from the IdP and pass this token in the initial connect call/message, provide the HLA SOM in advance, and support the ability to configure the federate name, federate type, and federation to join.

For the IEEE HLA 4 standard an extension to the Authorizer API is proposed, adding a method to authorize publish/subscribe operations. For the SISO WebLVC standard the addition of credentials to the connect message is proposed. Both standards currently miss the requirement for encrypted protocol communication. In the light of security, the addition of this requirement is strongly recommended. Lastly, standardization of the type of IdP (OIDC) and access token claims is recommended.

These are only the first steps in increasing security for setups like this; more explorations to other security controls are needed to further harden cloud-based simulation environments and their connections. As example, security controls for (C2) mediation services, further validation on compliance with the HLA SOM, deeper inspection on the exchanged data, and monitoring and logging are possible next steps.

ACKNOWLEDGEMENTS

This paper is developed by the TNO Cloud Based Modelling and Simulation research programme V2103, sponsored by the Dutch MoD. The authors would like to thank the CWIX partners for participating in the security tests at CWIX.

REFERENCES

Cybersecurity Division, C. (2023). *Zero Trust Maturity Model.* USA Cybersecurity and Infrastructure Security Agency.

IABG. (2024). *Simulation & Training systems*. Retrieved from https://www.iabg.de/en/business-fields/defence-security/services-solutions/smart-tools/simulation-training-systems

IEEE 1516. (2010). *IEEE Standard for Modeling and Simulation (M&S) - High Level Architecture (HLA) (IEEE 1516).* IEEE.

MASA. (2024). Retrieved from https://www.masasim.com/en/sword

Moller. (2022). HLA 4 Federate Protocol - Requirenents and Solutions (paper #24). *SISO Winter SIW.*

NATO. (2017). *Modelling and Simulation as a Service: Technical Reference Architecture.* NATO MSG-136.

NATO ACT. (2024). *NATO ACT Federated Interoperability*. Retrieved from https://www.act.nato.int/activities/federated-interoperability/

NATO DST. (2022). *STANREC 4799 NATO Reference Architecture for Distributed Synthetic Training.* NATO Standardization Office (NSO).

NATO FMN. (2024). *Federated Mission Networking*. Retrieved from https://www.act.nato.int/activities/federated-mission-networking/

NATO MSG-201. (2022). *NATO STO-Activities: Modelling and Simulation in Federated Mission Networking (FMN).*

NATO STANAG 4603. (2015). *M&S Architecture Standards for Technical Interoperability: HLA (STANAG 4603).* NSO.

NATO STO. (2024). *NATO STO*. Retrieved from https://www.sto.nato.int/Pages/modelling-and-simulation.aspx

OneSAF. (2024). Retrieved from https://www.peostri.army.mil/onesaf

OpenID Specifications. (n.d.). Retrieved from https://openid.net/developers/specs/

Pitch. (2024). Retrieved from https://pitchtechnologies.com/prti/

Rose, S., Brochert, O., Mitchell, S., & Connelly, S. (2020). *SP 800-207 Zero Trust Architecture.* NIST.

Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., & McQuaid, R. (2021). *NIST SP 800-160 Vol.2 - Developing Cyber-Resilient Systems.* NIST.

SAML Standards. (n.d.). Retrieved from https://www.oasis-open.org/standards/

Simlythics. (2024). Retrieved from http://simlytics.cloud/

SISO. (2022). *Standard for Web Live, Virtual, Constructive Protocol (SISO-STD-017-2022).* SISO.

USA DoD. (2022). *Zero Trust Reference Architecture.* USA Department of Defense.