A Learning-Based Zero-Trust Architecture for 6G and Future Networks

Conference Paper · September 2022

DOI: 10.1109/FNWF55208.2022.00020

Michael Enright
Quantum Dimension, Inc.

Eman Hammad
Texas A&M University – Commerce

Ashutosh Dutta
Johns Hopkins University

A Learning-Based Zero-Trust Architecture for 6G and Future Networks

Michael A. Enright
Quantum Dimension, Inc.
Huntington Beach, California, USA
menright@qdimension.com

Eman Hammad
Computer Science & Engineering
Texas A&M University - Commerce, RELLIS
College Station, Texas, USA
eman.hammad@rellis.tamus.edu

Ashutosh Dutta
Applied Physics Laboratory
Johns Hopkins University
Laurel, Maryland, USA
Ashutosh.Dutta@jhuapl.edu

***Abstract*—6G and Future Networks will require a dynamic, flexible, learning-based security architecture that will have the ability to handle both current and future cybersecurity threats. A distributed learning framework which can establish trust is needed that will enable technologies to be developed and integrated to meet security needs; one example of a distributed learning paradigm is Federated Learning. By integrating advanced learning with real-time digital forensics, e.g. monitoring compute and storage resources, it will be possible to develop a learning-based, real-time Zero-Trust Architecture (ZTA) that is necessary to achieve the highest level of security. With our proposed architecture, new techniques and machine learning techniques can be developed for enhanced real-time, adaptive and proactive security.**

***Index Terms*—machine learning, distributed machine learning, federated learning, cybersecurity, privacy, zero trust architecture**

I. Introduction

With the emergence and evolution in virtualization and softwarization motivated by cloud computing, a new networking paradigm, Software-Defined Networking (SDN) [1], was developed to address the need for greater network flexibility and efficiency. SDN fundamentally partitioned network traffic into three different planes: management, control and data. Similarly, the Cloud Security Alliance (CSA) developed and introduced the concept of Software-Defined Perimeter (SDP) motivated by the need for dynamic security in evolving network architectures [2]. SDP is an SDN-like architecture that can be applied to cloud security and other network architectures, which utilizes a controller to manage the security processes between hosts and clients similar to how SDN utilizes a management plane. The main theme is architectural flexibility, which presents major challenges for traditional security, further highlighting the critical need for a security architecture that could address such challenges. Most recently, the Zero-Trust (ZT) architecture concept emerged to holistically address the objectives of effective and efficient security for evolving and dynamic network architectures such as 6G and future networks.

Significant work has been undertaken by CSA, National Institute of Standards and Technology (NIST), European Union Agency for Cybersecurity (ENISA), and others, to develop secure network architectures. A security architecture, considers fundamental security capabilities on a system level and defines integration requirements, workflows and performance and reporting metrics. Security capability, that are similar to capabilities provided in advanced cloud platforms, include Identity and Access Management (IAM), key management, DevSecOps' continuous integration/continuous deployment (CI/CD), anomaly detection, etc. While such capabilities are crucial for advanced 5G and 6G networks security, more research is needed to arrive at a security architecture that focuses on the integration of learning and autonomy, specifically with Artificial Intelligence and Machine Learning (AI/ML) elements. Recalling that AI/ML is enabling 6G core functionalities and applications, then successfully arriving at a security architecture that clearly recognized the relevant complexities (layers/components/interactions), defines the guidelines and requirements for integration, and identifies performance metrics is of the highest priority. If successful, then a real-time, dynamic cyber-secure network would be more possible [3].

Federated Learning (FL) is a learning technique whose objective is to learn across multiple processing nodes in order to split an AI/ML workload. Essentially, it is a distributed computing paradigm with objectives that are similar to split and merge techniques. The promising applications of distributed autonomous network architectures supporting FL, where network cooperation extends to sharing modular duties, further highlights the critical and essential need for an applicable security architecture framework. In addition to the FL network processing the use-case application data, internally, it must attain the highest level of security to ensure that the learning network is safe and trusted. The objective of this work is to lay the foundation for a framework that can address this need by employing and aide in building trusted ZTA capabilities.

To further illustrate the challenges in scope of this work, we utilize the use-cases of edge computing and massive machine type communication (mMTC) in 5G/B5G/6G networks. These use-cases utilize AI/ML across several layers and parts of the network. If we further consider a specific critical application within these use-cases such as a public-safety application, then it can be understood how critical are dynamic and real-time security and resilience. Specifically, to enable an advanced security capability such as predictive security, then the system architecture should support the capability to ingest, process and act based on large-scale monitoring data. The architecture should also support sharing monitoring and security data across the network to enable real-time continuous threat modeling and response via AI/ML. Moreover, such capabilities need to extend to monitoring of AI/ML ecosystems themselves. In the absence of built-in trust and security in future networks' distributed AI/ML models, reliable use-cases such as edge and mMTC would remain largely infeasible.

AI/ML systems evolution has been intuitively led to be operated in simplified governance models, thus potentially creating more challenges and risks. For example, single entities such as companies, universities and research institutions, tend to have stove-piped design where the ML models, training data, and data pipeline platforms are under the control of one entity and often hosted in one location or service provider and are vulnerable to attack. While this model catered for earlier development stages, it does not reflect the practical use-cases leveraging service-oriented architecture where AI/ML models and data-science stages could be provided as a service. In the absence of strong and trusted governance, AI/ML models can be subject to attack [4]. Hence, security needs to be adaptive and the models updated securely.

The main objective of this article is to develop a framework where the security techniques can be integrated and coordinated to enable 5G/6G advanced and trusted dynamic and predictive cybersecurity, thereby supporting the fundamental objectives of ZTA. This is a fundamental step for future networks, specially given the factors of high-density network data and complex multi-step attacks. In [5], an attempt was made to estimate the timing of infected hosts. A comprehensive ML approach was developed in [6] by working with data from an enterprise Security Operations Center. A tremendous amount of data had to be processed which could not be performed in real-time for their ML model. This paper illustrates the challenges of developing a real-time security system. However, with computing capability and network cooperation, secure FL can be achieved to protect the network.

A. Organization and Contributions

The objective of this work is to propose a learning-based architecture, using techniques such as FL, to guide secure and trusted implementations of ZTA across 5G/6G and future network systems. This will support advanced security capabilities of 5G/6G networks, such as predictive security and security policy as defined by 3GPP security requirements [7]. The proposed architecture adopts the ZTA tenants as it unpacks related elements, layers and interfaces. In essence, this work is intended to support new and advanced, secure AI/ML algorithms and technologies across an array of use cases.

In Section II, we introduce future network security. In Section III, we address threat modeling and analysis for 5G and future networks. Section IV describes our secure, federated architecture. Finally, future development that includes risks and gaps are presented in Section V.

II. Future Networks' Security

Network cooperation continues to provide more capabilities and opportunities to address challenges in comparison with non-cooperative and autonomous networks particularly in resilience and reliability. Benefits of cooperation in networks have been shown to be advantageous specially considering examples such as the Internet, mobile ad-hoc networks (MANETs), cooperative navigation networks [8]. Extending this cooperative perspective to AI/ML, we can expect the powerful advantages of distributed computing in FL.

Benefits of FL in future networks are being examined for a wide array of different applications including improving communication efficiency [9], resource allocation and cross-layer optimization and others. However, a critical challenge in future networks regardless of the application is cybersecurity. As the connectivity and network aperture expands, so does the attack surface, and simultaneously threat attackers' capabilities and sophistication evolve as well. Consequently, it is most critical to have a dynamic and proactive security component as a core element in the cooperative network architecture, one that has the ability to evolve over time to counter existing and future threats. Equally as important is the fact that this security architecture must be able to adapt its operation mode (autonomous, semi-autonomous, collaborative) in order to mitigate security threats efficiently and effectively.

AI/ML is generally being increasingly utilized in future networks' cybersecurity detection and protection due to their desirable ability to continuously learn and adapt [10]. This is motivated by advanced cybersecurity objectives of being able to successfully address known (past and current) and unknown (future) threats particularly zero-day attacks. Further, FL specifically is proving to be a promising tool for advanced cybersecurity research. A good summary of challenges and future directions of FL for intrusion detection systems (IDS) can be found in [11]. Recent attention to security concerns in AI/ML deployments specifically for future networks aimed to better understand the security threats and propose solutions to address challenges via elements of security-by-design as evident in privacy-preserving algorithms. However, much of such efforts did not leverage a consistent architecture view and considered silo-ed elements in their analysis.

In this work, we focus on the future networks' security as a use-case driven by the unique differentiating requirements such as real-time, dynamic, and predictive capabilities. For example a dynamic predictive security will capitalize on innovative approaches to real-time digital forensics to continuously train and adapt [12]. It is important here to note that the proposed architecture can be applied to other use-cases.

An example end-to-end architecture of a 5G network with the overlay of security threats is illustrated in 1 [13]. As can be observed from the figure, security threats can target any part of the network including end devices (e.g. IoT), edge platforms, radio access networks (RAN), transport, clouds, core networks, SDN, open systems/interfaces, and several more logical and physical components. With virtualization and

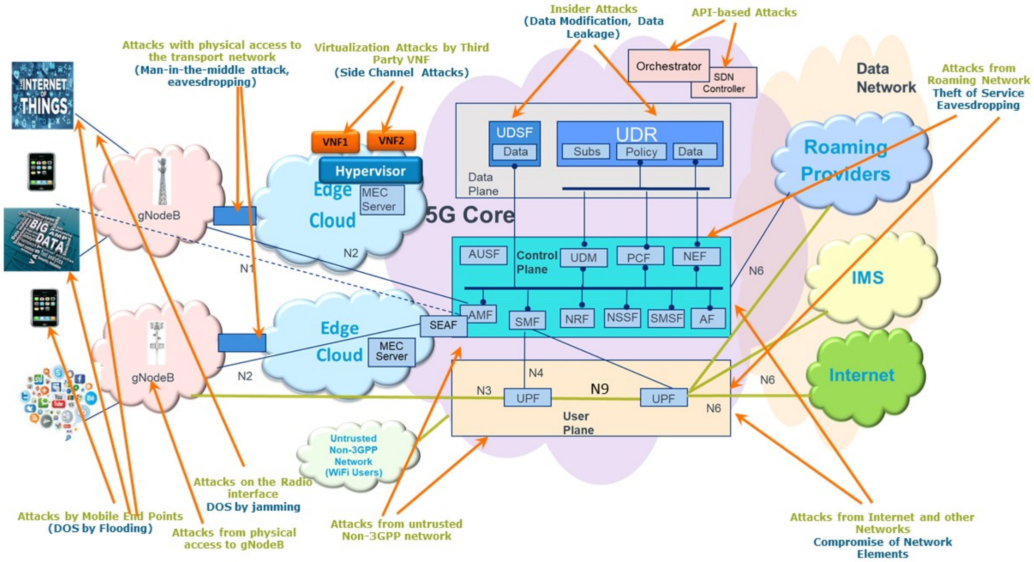

Insider Attacks
(Data Modification, Data Leakage)
API-based Attacks
Attacks with physical access to the transport network
(Man-in-the-middle attack, eavesdropping)
Virtualization Attacks by Third Party VNF
(Side Channel Attacks)
Orchestrator
SDN Controller
Data Network
Attacks from Roaming Network
Theft of Service
Eavesdropping
INTERNET OF THINGS
BIG DATA
VNF1
VNF2
Hypervisor
Edge Cloud
MEC Server
UDSF
Data
UDR
Subs
Policy
Data
Data Plane
5G Core
Roaming Providers
gNodeB
N1
N2
AUSF
Control Plane
UDM
PCF
NEF
N6
IMS
SEAF
AMF
SMF
NRF
NSSF
SMSF
AF
Edge Cloud
MEC Server
gNodeB
N2
N4
N3
UPF
N9
UPF
N6
N6
Internet
User Plane
Untrusted Non-3GPP Network (WiFi Users)
Attacks on the Radio interface
DOS by jamming
Attacks by Mobile End Points
(DOS by Flooding)
Attacks from physical access to gNodeB
Attacks from untrusted Non-3GPP network
Attacks from Internet and other Networks
Compromise of Network Elements

Fig. 1. Attack surface of a 5G network [13].

open interfaces, the attack surface is expanded to include many more networked devices and services, hence greatly increasing the associated security risk. For example, open systems, such as Open RAN (O-RAN) and Cloud (C-RAN), have exposed the internal workings of the radio architecture presenting an opportunity to be exploited by potential attackers. Any device that connects to the network is a potential source of a vulnerability, and with such a wide breadth of potential threat vectors, security intelligence and autonomy are crucial.

5G threat landscape underscores the critical need for security approaches that are: 1) autonomous, 2) distributed, 3) dynamic, 4) predictive, 5) integrated by design, 6) widely applicable, and 7) provide defense-in-depth. And more importantly, an architecture that enables such approaches and components to operate effectively, securely and efficiently. Autonomy is critical for secure operation of future networks, however, autonomy can be exploited in the absence of adequate frameworks and architectures. It is imperative here to highlight the complexity of addressing security while ensuring that the infrastructure supporting security is secure and trusted itself. Hence, the importance of adopting an architecture that provides provisions across the logical layers to address this complexity.

The proposed security architecture must be able to both orchestrate and monitor security capabilities. This will require a cooperative system to disseminate, process and update critical information that include current security threat vectors and levels, changes in AI/ML models and associated parameters and other inputs. The result will be a dynamic system architecture where real-time security situational awareness can be achieved, thereby necessitating real-time management, or orchestration, in order to operate efficiently and effectively against current and future threats. Some of the core elements of such a system are listed below:

- **Device and Edge Platform Security Functions** – Must mitigate threats originating from a local area or open interface reducing the risk of such threats propagating to the rest of the network.
- **Network Security Functions** – Must detect and mitigate threats to the network across the network layers, zones and applications.
- **Supervised and Unsupervised AI/ML Algorithms** – Must provide and facilitate an ecosystem that allows the use of most if not all AI/ML models as applicable for security functions.
- **Open Interfaces** – Interfaces must be specified so that new technologies can be implemented seamlessly and will provide real-time situational awareness.
- **Threat Vector Sharing** – Must allow and facilitate the sharing of threat vector information in real-time with other models and system components as needed.
- **Online Training and Live Updates** – Must allow for trusted AI/ML online training and model updates.
- **Dynamic Model Generation** – Must allow for autonomous capabilities such as generating and deploying new model AI/ML models in response to system context and threat analysis.
- **AI/ML Security Orchestration** – Must facilitate coordination between all elements of the AI/ML ecosystem.
- **Monitoring** - Must allow for collection of monitoring data and events for security detection and response functions.

The next step is to ensure that the system itself is resilient against attacks by using a framework that addresses important security tenets.

A. Zero Trust Architecture

Zero-trust (ZT) was recently promoted [3] as a promising approach that in principle presents a shift in complex systems security. ZT acknowledges slowly eroding perimeters and a wider threat landscape comprising external and internal threat actors. ZT frameworks adopt three key principles; to always verify explicitly, grant access to resources on least privilege basis, and to assume breach. By means of those principles, ZT puts very little assumptions (if any) that could often cause gaps in current traditional security approaches. For example, traditional approaches seek in effect to white-list or black-list active entities be it a user, application or flow; however assumptions such as known applications or trusted devices could lead to bypassing security controls resulting in systems' vulnerabilities to several threat vectors [14]. ZT highlights the need to always challenge those assumption as any entity could be exploited at any point in time. Hence, ZT implementations need to be thought through carefully to enable effective remediation and mitigation combining capabilities for authentication, authorization, access control and active monitoring [3], [14].

Zero Trust is an architectural concept and philosophy that does not specifically outlines implementations. Hence, there will be different ways to implement ZT within a network. We focus on 6G and future networks to present an architecture than can create a zero trust implementation through learning and autonomy. Dynamic trust should rely on situational awareness by continuously evaluating entities within the network and their associated security state. The goal should be a *learned trust* that is gained by continuous monitoring and learning through AI/ML techniques of today and in the future.

However, security of AI/ML systems presents a unique challenge because of the complexity, sensitivity and particular nature of its sub-components as described in [4]. Future networks are one example of critical systems that will heavily rely on FL to both 1) operate the systems at a high level of efficiency, such as through network autonomy, and to 2) enable its self-adaptive real-time security capabilities limit both the number and magnitude of threats. This motivates investigating how a secure-by-design approach can be developed for FL taking into account ZT principles thereby enabling a unified and structured approach. Equally important in this endeavor is establishing the required architectural components essential to enable this treatment and inform possible implementations. The intent of this work is not to define a stringent set of requirements, thereby limiting design choices. Instead, we propose a dynamic architecture that has the structure to enable advanced FL capabilities across many types of complex networks.

III. Threat Modeling & Analysis

In this section, we expand on the cybersecurity challenges of distributed AI/ML. This understanding is essential to enable subsequent discussions in this paper. A distributed AI/ML algorithm can be viewed as a multi-phase system where data is used to train a model that's part of a production architecture [15]. With this perspective, we can proceed to describe threats and relevant security controls as it relates to: 1) data, 2) model, and 3) architecture within the machine learning life-cycle.

A. Threat Taxonomy

Hence, we can summarize the main security challenges in this context as follows:

- **Infrastructure Hardware and Software Security** - Vulnerabilities in hardware, platforms and applications can be exploited by threat actors to gain access and manipulate the data and/or models.
- **Data Integrity** - Adversarial threat actors can inject malicious data in the training stage to affect the inference capability of AI models or add a small perturbation to input samples in the inference stage to change the inference result [15].
- **Data Privacy** - In applications where users provide their data, an adversarial threat actor can repeatedly query a trained model or intercept data exchange between the distributed learning components to infer private information.
- **Model Confidentiality** - A knowledgeable adversary may be able to create a clone model using inferences obtained through a number of queries against the original model.
- **Model Security and Robustness** - In a FL network, model parameters are updated and shared across nodes. Model updates must be done in a secure manner. In addition, the model must be secure against adversarial attacks, such as those current being developed using Generative Adversarial Learning (GAN) [16], [17] concepts.

Given such challenges, a ZT approach for FL requires a clear definition of what comprises an entity and a distinction of applicable threats to each entity type in addition to applicable mitigation approaches that this paper aims to generalize. Based on the standard ML life-cycle we can decompose FL into three entity types 1) data, 2) model and 3) architecture. The combination of those entities will cater to the different classes of applications and their requirements. However, this will remain helpful as we traverse the threat vectors against each entity.

Data can be targeted through manipulation. This can be characterized by different attack vectors. In evasion attacks, carefully designed variations of input data are used to drive the model outcomes away from the main objective. For example, to cause a model to overlook anomalies. In poisoning attacks, mixing a small percentage of manipulated data with the rest of the data can be used to significantly impact the model accuracy. Finally, data can be a target by attack vector targeting confidentiality and privacy. Current approaches to mitigate those threats utilize data or algorithms. Data can be augmented with adversarial data samples to train the model on detecting and handling evasion attacks. Similarly, filtering and regression can be utilized to address poisoning attacks, and finally algorithms such as differential privacy can be leveraged to defend against data breach threats.

Models can be targeted in inference attacks to extract the trained model constituting intellectual property theft, and could further used to generate data that can be leveraged against the original model. Several references aimed to discuss the complexity of potential defenses given the blockbox nature of some of the models. Most recent works have aimed to tackle this through the developments of enhanced model security. Model security can be improved through model 1) detectability, 2) verifiability, and 3) explainability.

Architectures of FL exposes it to a multitude of threat vectors that can be mapped to threat vectors against distributed systems. Threat actors would aim towards not only data and model vulnerabilities by also against the architecture. For example, if the FL model has a centralized global entity, it could be targeted to decimate decision making or shed doubts on the global situational awareness.

B. Example Risk Scenarios

We next discuss a few example risk scenarios based on 5G use cases with FL implementations. This is will be considered for two use-cases 1) network security function virtualization for massive Machine Type Communications (mMTC), and 2) operational intelligence to support Ultra Reliable Low Latency Communication (URLLC) applications.

- **mMTC Applications** - The 5G mMTC model supports a large number of low data rate sensors that can be on the order of millions. These can be employed in manufacturing and warehousing applications, smart city application such as road sensors, parking meters, etc. Because they are part of the network, each of these has the ability to attack the network. These can be used in distributed denial of service attacks (DDOS) and others. With these approach described in the following, these attacks can be mitigated by smart algorithms that reside on the edge network that communicates with these sensors or at the Radio Access Network (RAN) if the devices communicate directly two it.
- **URLLC Applications** - This model requires resilient, assured delivery since these applications include medical, public services, etc. Disruption of network security by using jamming, DDOS, man-in-the-middle and other attacks can have grave circumstances for users of these services. Hence, recognition of the network security state is paramount to routing these services around disruption points.

IV. Learned Trust Architecture for Cybersecurity

6G and Future Networks will have many different model types of AI/ML applications that support network management and optimization, security policy, channel optimization and more. As described in II-A, a ZTA can be used to support secure FL for these applications. However, the ZTA that we describe here is a dynamic one that can learn trust over time rather than a static one.

Beginning with the distributed aspect of cloud computing and the management and control aspect of SDN, we propose an architecture, in terms of layers and components, that is suitable for learning trust. This may seem counter to ZT, where one may assume that no one is trusted. In fact, trust is established through fixed ”relationships”, such as shared keys. However, entities may be compromised, so trust should be dynamic and determined based upon and operational state or situational awareness. This is especially true in FL where there many be many different sources of information that contribute to the ML model.

A. Layered View

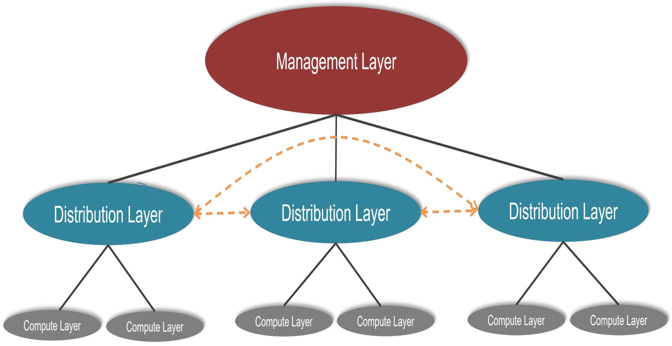


Fig. 2. Layered view of the learning-based architecture.

The architecture is described in terms of layers, as shown in Fig. 2, and components, as show in Fig. 3. The layers illustrated in the figure demonstrate the system-level functionality, i.e. how the system operates as a whole rather that functions that are assigned to different entities within the future network. For example, there is a management component at the control node but the same component does not exist at client nodes. In this way, he Management Layer is similar to a control node in SDN.

The ***Management Layer*** is responsible for for network management and security. Some responsibilities of network management in a FL environment include:

- Adding and removing nodes from the network
- Maintaining lists and their locations of AI/ML data files, models etc.
- Manages AI/ML model training and updates
- Coordinating traffic among network nodes
- Ensures ”authenticity” and ”validity” of model updates

The security aspect of the Management Layer has different responsibilities that are split across the control and client nodes which includes:

- Orchestration of security functions throughout the network
- Contains AI/ML models for different security functions, such as IDS, physical layer security, compute layer security, etc.
- Receives status and alerts from nodes in real-time

- Controls and updates security policy based upon network status

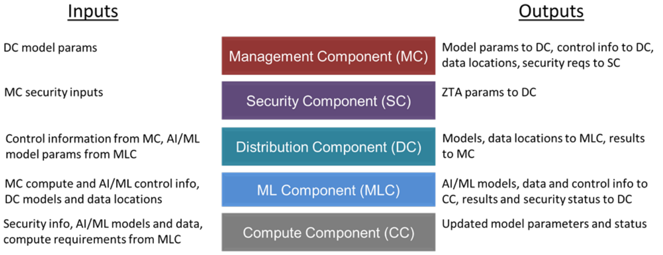
In order to minimize traffic and compute requirements at the Management Layer, this layer is primarily responsible for AI/ML and security orchestration. The Management Layer uses the Distribution Layer to disseminate and receive messages from the nodes. From the component perspective, this layer can be seen as implementing the Management Component and Security Component, although not in its entirety.

As its name implies, the ***Distribution Layer*** is responsible for the distribution of messages and data between different network nodes. This is especially important for FL, since training data may reside at different nodes in the network and may need to be transferred as such. Hence, the Distribution Layer can be seen as the freeway that connects the different elements and manages traffic. This layer also implements the security policy as defined by the Management Layer. The Distribution Layer implements functionality from the Distribution Component and the Security Component.

From Fig. 2, this layer connects Compute Layer, which does all for the AI/ML processing on its or cluster of nodes. The figure shows only to Compute Layers, but this is not a limit.

Finally, the ***Compute Layer*** is responsible for the computational processing associated with implementing and training the models. It is envisioned that there are many AI/ML models running on a particular node or set of nodes. These could be NN models, LSTM, unsupervised models, etc. This layer receives models and data from the Distribution Layer and provides its results to that layer, which are subsequently forwarded to the Management Layer.

B. Component View



Inputs		Outputs
DC model params	Management Component (MC)	Model params to DC, control info to DC, data locations, security reqs to SC
MC security inputs	Security Component (SC)	ZTA params to DC
Control information from MC, AI/ML model params from MLC	Distribution Component (DC)	Models, data locations to MLC, results to MC
MC compute and AI/ML control info, DC models and data locations	ML Component (MLC)	AI/ML models, data and control info to CC, results and security status to DC
Security info, AI/ML models and data, compute requirements from MLC	Compute Component (CC)	Updated model parameters and status

Fig. 3. Component view of the learning-based architecture.

To support learned trust, there are five different components, as illustrated in Fig. 3: Management Component (MC), Security Component (SC), Distribution Component (DC), Machine Learning Component (MLC) and Compute Component. In this way, the distributed CC functionality can be partitioned by functionality. For example, the AI/ML algorithm implementation and training can be kept separate from the management and security of the network. Furthermore, these components do not necessarily need to be resident at the control and client nodes, which is illustrated in Fig. 4.

The ***Management Component*** component is one component of the Management Layer. It is responsible for operating the FL network by orchestrating the AI/ML process by collecting model coefficients and architectures, security parameters, etc. and using its distribution component to distribute these to network nodes. It is also responsible for orchestrating security policy. This layer manages the functions that are implemented by the SC and DC at the control node.

The ***Security Component*** takes security state information from the MC and creates the security policy and functions. It may use advanced methods such as ML models derived from data received from the network elements. As such, this is the powerful security engine that manages security across the network.

The ***Distribution Component*** has two incarnations. The first resides on the control node and manages the distribution of data across the network. The second sits and the client node and communicates model and training data with other nodes. While control information is passes from the control node, communication between nodes is direct to minimize network traffic.

The ***Machine Learning Component*** is responsible for the model implementation and all AI/ML related functions such as continuous monitoring, integration of multiple models, model training, etc. It interfaces with the network through the DC and processing is performed by the CC.

The ***Compute Component*** performs all computations needed for model processing and training. It runs the models from the MLC, but it may also gather parameters from the node's operating system regarding potential attacks. Thus, in addition to running models, the CC is responsible for collecting security, or even sensor data, and passing it up to the MLC.

A brief summary of each layer is provided in Table I.

TABLE I
COMPONENT DESCRIPTIONS.

Component	Description
Management	AI/ML Network management and control, similar to ITU MLO
Security	Secures models and contains sandboxing for verification and validation to ensure integrity "accurate, tamper-free" models are integrated into the ML layer, verification of data sources/authentication, trust platform, as in multi-agent platforms
Distribution	Trusted network formation and model parameters, training data distribution
Machine Learning	Hosts AI/ML algorithms of different types (supervised, unsupervised, etc.), integrates these together, maintains specified interfaces and contains model implementations
Compute	Focuses on distributed computing and virtualization, Apache Spark is an option

C. Network Topology, Software Architecture and Use Cases

The aforementioned architecture described thus far is not limited to any particular network topology - centralized, decentralized, hierarchical multi-layer, etc. However, as applied to a 5G/B5G type of network, a centralized topology is in line with those found in practice, or carrier networks. It is also in line with the framework set forth by ETSI for MLFO. Thus, that is the approach that we take in this paper, but obviously, a decentralized approach such as found in blockchain, with cryptocurrency being a prime example, is a very interesting and new area related to FL and it will be an area that we will focus on in the future.

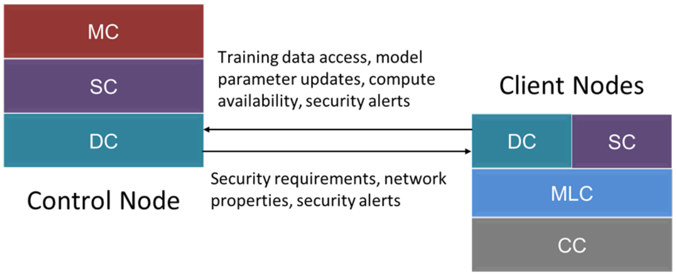

MC
SC
DC
Control Node
Training data access, model parameter updates, compute availability, security alerts
Security requirements, network properties, security alerts
Client Nodes
DC
SC
MLC
CC

Fig. 4. Information transfer between control and client nodes.

A centralized model is shown pictorially in Fig. 4. The figure is simple and meant to demonstrate the interaction between the control node and client nodes and is a mixture of our layer and component perspectives. The duties of the control node are centered on management of network operations. This also includes security policy across the network. For example, the MC maintains the ”objects” and databases that are needed for FL, security state, and others. In the case of FL, the MC know the locations of all training data, AI/ML models that are being used, and the security state at each node. Implementation of this policy resides with control node's SC and distribution takes place via the control node's DC. Essentially, this is a client-server architecture with the control node acting as a server, but the tasks are not trivial.

All intelligence in terms of network operation, security policy and control, and AI/ML orchestration reside a the client node. Performing these functions for a large network, as is the case in 5G/B5G, is no easy task. From a FL perspective, this is a very interesting problem. Consider the fact that there are many nodes with different sources of data that can be used to secure the network, plus the fact that this data is not sparse. TCP/IP data streams are dense in terms of computing requirements, but these are not the only sources of data. Real-time forensic data, e.g. processing load, file access attempts, and others, paint a picture of the security state of the network.

The client node processing is partitioned into three areas: distribution and security, AI/ML model implementation and training, and compute processing. Distribution and security are linked together due to the fact that secure links are required to share the data. For example, SSH can be used between nodes in the network. Additionally, the SC monitors the security state of the node by receiving security policy information from the control node. A second function of the SC is to collect security metrics, i.e. forensics, and send them to the control node. Essentially, it is monitoring the state of the node to ensure that there has been no intrusion, and hence, any training data or metric collected from the node are valid.

The CC is an importance resource in the network architecture. Not only is in critical to host environment; it can also be used to train models as part of the FL architecture. For example, consider the case of edge computing in a corporate or 5G/B5G environment. Depending the computing resources available, training can be allocated to compute elements in much the same was as water-filling to maximize channel throughput subject to an energy constraint as described above.

D. Zero Trust Architecture Support

Our architectural approach can be applied to different network topologies from enterprise to the Internet, the key point being that cooperation can lead to a ZTA by implementing cooperative algorithms that can address all levels of the security stack. Because of the nature of cyber attacks, a distributed, and federated, systems has the ability to detect cyber attacks more effectively. For example, if a node has been comprised, self-detection may not occur since a root-level attack may turn off all defenses. However, should the attacker attempt to communicate with other nodes, clearly, its presence will be detected.

In Table II, we describe how this architecture can be used to implement ZT using cooperation. The key takeaway from this table is that for a secure network, the following items, at a minimum, are required:

1) *Network Registry* - For a secure FL system, a registry of all nodes and resources in the network should reside at the MC. The registry database contains not only the current state of the network, but also the historical transactions, such as file access attempts, etc. This information can be used to set the network security policy via AI/ML methods.
2) *Trusted Implementation* - While SSH and encrypted tunnels can secure communication, they do not guarantee trust. It must be assumed that insider attacks will occur, so methods are needed to detect this by monitoring security analytics, file operations, network communication, etc.
3) *Real-Time Security Analytics* - Knowledge of the current security state of network nodes is essential to securing the network. Computing metrics, file access attempts, state of the key management system are all important elements in being able to determine whether a node has been compromised.

V. Future Development; Risks and Gaps

The fundamental view of a secure FL architecture is that computing capability should be a core component of the architecture. Rather than simply saying that each node can compute its own model coefficients, for example, we see the need to distribute data across that network such that other nodes can contribute to the compute processing. In doing so, this will create the possibility of a richer set of algorithms, not just from a run-time perspective but also from a training perspective. One such example is with IoT devices that may collect data but not have sufficient compute capability to update model coefficients.

TABLE II
ZERO-TRUST ARCHITECTURE USING LEARNING-BASED ARCHITECTURE.

Tenet	Architecture Support	Notes on Implementation
1. All data sources and computing services are considered resources.	The Management Layer monitors all resources in the network. Access to the network is controlled by the MC.	• Registry of nodes and elements of the network at MC • Knowledge of network node state is required • Need for AI/ML algorithms for dynamic security policy
2. All communication is secured regardless of network location.	The SCs within the Control and Client nodes use encrypted channels such as SSH, tunneling protocols, etc.	• Autonomous management of encryption keys and secure tunnels is required • Knowledge of node security state is needed
3. Access to individual enterprise resources is granted on a per-session basis.	The MC knows all resources in the network and distributes access rules to the client nodes via the Distribution Layer.	• Smart key management routines
4. Access to resources is determined by a dynamic security policy and may include other behavioral and environmental attributes.	The CC monitors computing performance, such as memory usage and system performance, and relays this to the Security Component at the Control Node for continuous monitoring. Similarly, the SC monitors resource access requests and reports these to the Control Node.	• Policy is set forth at MC • Security analytics are needed to derive policy • AI/ML algorithms are needed to derive security state from analytics
5. The enterprise monitors and measures integrity and security posture of owned and associated assets.	This is similar to Tenent #4 with real-time integrity monitoring.	• Security state and real-time analytics are inputs to this function
6. All resource authentication and authorization are dynamic and strictly enforced before access is allowed.	The Management Layer controls access to resources on a per-session basis.	• Must detect insider attack where key is valid but an attack is underway. • Trust through intelligent key management and state monitoring are required. • Security analytics and network communication monitoring may be able to detect insider attack.
7. The enterprise collects information about the current state of assets, network infrastructure and communications.	The same approach with real-time monitoring and centralized control as in Tenets #3 and #5 is employed here.	• This is the security analytics and monitoring that was described earlier.

With the aforementioned architecture and future technology developments, AI/ML-based security will be a key component of future networks. By developing the aforementioned technical approaches and frameworks into a suite of 5G and Future Network products, the great potential of AI/ML for security can be realized.

Ultimately, the goal is to develop the dynamic, predictive AI/ML system such that it can support advanced capabilities to secure Future Networks beyond what is capable today.

REFERENCES

[1] Wikipedia, "Software-Defined Networking," https://en.wikipedia.org/wiki/Software-defined_networking/, accessed: July 24, 2022.

[2] J. Garbis and J. Koilpillai, "Software defined perimeter architecture guide," *Cloud Security Alliance*, 2019.

[3] NIST, "Zero Trust Architecture," https://csrc.nist.gov/publications/detail/sp/800-207/final, accessed: July 24, 2022.

[4] ETSI GR SAI 004, "Securing Artificial Intelligence (SAI); Problem Statement," https://www.etsi.org/technologies/securing-artificial-intelligence, accessed: July 24, 2022.

[5] X. H. et. al., "BAYWATCH: Robust beaconing detection to identify infected hosts in large-scale enterprise networks," in *Proc. of the 46th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks*. Toulouse, France, July 2016, pp. 479–490.

[6] A. O. et. al., "MADE: Security analytics for enterprise threat detection," in *Proc. of the 34th Annual Computer Security Applications Conference*, Dec 2018, pp. 124–136.

[7] 3GPP TS 33.501, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security architecture and procedures for 5G system (Release 17)," vol. 17.2.1, Jun 2021.

[8] M. A. Enright and C. N. Kurby, "A signals of opportunity based cooperative navigation network," in *Proc. IEEE National Aerospace & Electronics Conference (NAECON)*. Dayton, OH, July 2009, pp. 213–218.

[9] J. K. et. al., "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2017.

[10] M. X. et. al., "Machine learning security: Threats, countermeasures, and evaluations," *IEEE Access*, vol. 8, pp. 74 720–74 742, 2020.

[11] S. A. et. al., "Federated learning for intrusion detection system: Concepts, challenges and future directions," *arXiv preprint arXiv:2106.09527*, 2021.

[12] S. V. N. Parasram, *Digital Forensics with Kali Linux*, 2nd ed. Birmingham-Mumbai: Packt, 2020.

[13] A. Dutta and E. Hammad, "International Network Generations Roadmap (INGR), Virtual Workshop, Security Working Group," June 2020.

[14] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3gpp 5g networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 170–195, 2019.

[15] Huawei, "AI Security White Paper," https://www-file.huawei.com/-/media/corporate/pdf/cyber-security/ai-security-white-paper-en.pdf, accessed: August 24, 2021.

[16] J. S. I. Goodfellow and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[17] A. C. et. al., "Generative adversarial networks: An overview," *arXiv preprint arXiv:1710.07035*, 2014.