TLP:CLEAR

DOT/FAA/TC-693KA8-23-D-00190/TO 26

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Part 3 Cybersecurity Data Science (CSDS) Aviation Architecture Framework (AAF) - Implementation Guidance

August 23, 2024

Version 1.2

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report. The findings and conclusions in this report are those of the author(s) and do not necessarily represent the views of the funding agency. This document does not constitute FAA policy. Consult the FAA sponsoring organization listed on the Technical Documentation page as to its use.

This report may be made available upon request to the FAA Aviation Research Division.

Technical Report Documentation Page

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

1. Report No.

DOT/FAA/TC-693KA8-23-F-00190/TO26

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Title of Report: Cyber Security Data Science- Aviation Architecture Framework

Subtitle of Report: System Guidance Document

5. Report Date

August 2024

6. Performing Organization Code

7. Author(s)

Center for Aerospace Resilient Systems (CARS)

Jayson Clifford
Lauren Warner
Daniel Diessner
Dr. Ilhan Akbas
Isidore Venetos

8. Performing Organization Report No.

9. Performing Organization Name and Address

Embry-Riddle Aeronautical University
Center for Aerospace Resilient Systems (CARS)
1 Aerospace Blvd. Daytona Beach, FL 32114-3900

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

693KA8-23-F-00190/TO 26

12. Sponsoring Agency Name and Address

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

13. Type of Report and Period Covered

14. Sponsoring Agency Code

ANG-E271

15. Supplementary Notes

16. Abstract

This document is the third part of a series of four (4) core documents to provide an overview of the top-down output from the FAA Cybersecurity Data Science Aviation Architecture Framework (CSDS AAF) research program. The intent of this document is to be used as a general guidance reference for the implementation of CSDS AAF in various systems across the aviation ecosystem and to be used as a reference for industry standards or guidance activities. The four (4) core CSDS AAF documents are:

- **Part 1 CSDS AAF - Overview & Value Proposition**: The primary purpose is to communicate to aviation stakeholders the vision and potential value of the FAA CSDS research and generally how it could potentially be leveraged to address key aviation cybersecurity challenges.
- **Part 2 CSDS AAF - Technical Specification Document**: As an ontology for the CSDS Aviation Architecture Framework, this document provides narrative to describe and explain all of the key AAF components and functions, coupled with diagrams to illustrate the overall AAF structure.
- **Part 3 CSDS AAF - System Guidance Document**: This document provides guidance for implementation of the CSDS AAF which is defined in the AAF Technical Specification Document.
- **Part 4 CSDS AAF – Glossary & Acronyms:** This document provides the Glossary and Acronym material for all parts of the CSDS AAF documents.

17. Key Words

Security Data Science (CSDS), Aviation Architecture Framework (AAF), Guidelines, Cyber Analytical Capability, Data Life Cycle, Interconnected Individual Systems

18. Distribution Statement

This report may be made available upon request to the FAA Aviation Research Division.

19. Security Classif. (of this report)

Unclassified

20. Security Classif. (of this page)

Unclassified

21. No. of Pages

22. Price

Contents

Figures

Tables

No table of figures entries found.

Acronyms

See Part 4: Cybersecurity Data Science (CSDS) Aviation Architecture Framework (AAF) – Glossary & Acronyms.

Executive summary

A critical challenge in cybersecurity is determining if a cyber incident has or is happening. Data science holds promise to more quickly and more effectively find anomalous data that could indicate a cyber incident. The Cybersecurity Data Science (CSDS) Aviation Architecture Framework (AAF) seeks to apply data science to the Aviation Ecosystem, which involves both Information Technology (IT) and Operation Technology (OT) with various stakeholders such as airlines, airports, and Original Equipment Manufacturers (OEMs). This document defines the conceptual elements and taxonomy of the CSDS AAF. The CSDS AAF Systems Architecture applies this framework in different Environments of Operation and involves Stakeholder Data-Stores, Cyber Analytical Capabilities (CAC), and Interconnected Individual Systems (IIS). The framework also introduces the CSDS AAF Data Life Cycle, which consists of Acquire, Pre-Analyzed, Collect, Advanced Analytics, and Information Sharing. A critical component in this data perspective is collecting the appropriate data which is described using the Data Sphere concept. The document then describes how to apply and refine the CSDS AAF for a specific Environment of Operation.

1 Introduction

This is the third part of a series of four (4) documents to provide an overview of the top-down output from the FAA Cybersecurity Data Science Aviation Architecture Framework (CSDS AAF) research program. The intent of this document is to be used as a general guidance reference for the implementation of CSDS AAF in various systems across the aviation ecosystem and to be used as a reference for industry standards or guidance activities. The four (4) core CSDS AAF documents are:

- **CSDS AAF – Part 1: Overview & Value Proposition**: The primary purpose is to communicate to aviation stakeholders the vision and potential value of the FAA CSDS research and generally how it could potentially be leveraged to address key aviation cybersecurity challenges (Embry-Riddle Aeronautical University Center for Aerospace Resilient Systems, 2024).
- **CSDS AAF – Part 2: Technical Definition**: As an ontology for the CSDS Aviation Architecture Framework, this document provides a narrative to describe and explain all of the key AAF components and functions, coupled with diagrams to illustrate the overall AAF structure (Embry-Riddle Aeronautical University Center for Aerospace Resilient Systems, 2024).
- **CSDS AAF – Part 3: Implementation Guidance**: This document provides guidance for the implementation of the CSDS AAF, which is defined in the AAF Technical Definition (Embry-Riddle Aeronautical University Center for Aerospace Resilient Systems, 2024).
- **CSDS AAF – Part 4: Glossary & Acronyms**: This document provides the Glossary and Acronym material for all parts of the CSDS AAF documentation (Embry-Riddle Aeronautical University Center for Aerospace Resilient Systems, 2024).

2 Cybersecurity Data Science Aviation Architecture Framework (CSDS AAF)

The structure of Parts 1 and 2 of the CSDS AAF documents has been replicated in Part 3. All three (3) parts are intended to provide the complete definition of the CSDS AAF. Some sections in Part 3 may not contain any information if it has already been sufficiently described in Part 2: Technical Definition. In those cases, the section headings are left in place to maintain the structure across all three parts of the document, and the previous part of the document will be referenced.

2.1 CSDS AAF Conceptual Elements

The conceptual elements can be found in Part 2.

2.2 CSDS AAF Taxonomy & Reference Model

The taxonomy and reference model definitions can be found in Part 2.

2.3 CSDS AAF Systems Architecture: The Systems Perspective

This section will provide guidance statements for implementing the various systems within the CSDS AAF. The information in these sections is intended to be generic to any environment of operation or use case. Most sections below are organized into three parts, each focusing on a particular aspect of the system: functions, performance, and security. Some sections may not contain guidance on a specific topic if it is not applicable or if it has already been sufficiently described in Part 1: Overview & Value Proposition and Part 2: Technical Definition.

2.3.1 AAF Operational Concept (System Architecture)

The AAF operational scenario and concept can be found in Part 1 and Part 2 respectively.

2.3.2 System Architecture Aspects of Environments of Operations

2.3.2.1 Interconnected Individual Systems (IIS)

The IIS is a foundational element connected to several architectural components of the CSDS AAF, such as data acquisition sensors, local storage, and data egress points. When answering the three key CSDS objective questions – Is there a cyber-event pending? Is there an attack occurring now? Was an incident/event caused by cyber activity? – the object of the question will often be an IIS. Data acquisition sensors (DAS) can be placed in the IIS to collect, filter, and pre-analyze data. The data is placed in local storage and sent via specialized data egress points to one or more Data-Stores within the environment of operations. This data reflects the current and past state of components within the IIS. The data is used in the CSDS AAF as input to the Cyber-Analytical Capability (CAC).

2.3.2.1.1 Functions

What are the functions of an IIS with respect to the AAF? As defined by the AAF taxonomy, it is the basic building block within any particular environment of operation. It's a foundational component capable of capturing, storing, filtering, and sending relevant data. It is recommended that the IIS and its components are inventoried and documented. Some considerations when documenting the IIS:

- If already documented, where is this information stored currently?
- How would the information reference the IIS and its components inside analytical models?
- Who is responsible for administering the components? Is there shared ownership of the data in/about the IIS between different stakeholders within the environment of operation?
- Do the data owners have sufficiently granular control and visibility into these systems to inform an analytical model?

This information should be accessible in some form to the CAC, ideally referenced in metadata by a globally unique identifier for each component (i.e., UUID, as defined in ISO 9834-8:2014 or IETF-4122) to allow for traceability. This identifier can be used as a reference in models and data structures linking all associated data to a particular IIS. Some examples of CSDS-relevant component documentation are listed below:

- Serial and model numbers of components, current firmware, operating system, VM hypervisor and/or iOS or similar software capability versions and configurations.
- Component owners and those responsible and accountable for administrating the components.
- The physical connections to the components inside and outside the IIS system boundary.
- Any known connection information such as interfaces, ports, protocols, etc.

The IIS can present attack surfaces that are of specific interest to CSDS:

- External connections to the IIS – systems controlled by a separate authority outside the environment of operation may maintain connections that bypass security controls to allow for remote access or updates by vendors or service providers.

2.3.2.1.2 Performance

With the primary driver being to decrease the dwell time for cyber events, how the relevant data is offloaded, retrieved, or streamed from an IIS has a large impact on the efficacy of the functions in the CAC. These systems are very diverse; in some cases, they may have a high throughput, low latency optical fiber connection, while others may only have access to low-power wireless or direct serial connections to a host PC. In the worst case, the data may need to be offloaded manually to removable storage, in some cases as archaic as a 3.5” floppy disk. Due to this, it is necessary to address the basic parameters of the connection and its position on the network to understand what kinds of data can be provided and how frequently it can be sent.

In addition to the benefits to defining analytical models, knowing this information can be useful when building up CSDS capability in the environment of operation since there may be some systems that arbitrarily, rather than necessarily, have poor availability with sporadic or low-bandwidth connectivity. A better option may be available that is not used because the lower-quality connection has met the need. But as the need for low-latency observability into these systems increases, the team may need to take advantage of those options if they are available and its practicable to activate and maintain them.

There is also a certain degree of feedback loop here with the analytics, as once everything is up and running, the analysts may determine that a particular model will only really work with higher availability of a data element that is not currently available or isn't coming in often enough to be useful.

Following that, is it recommended that every component be connected at all times with enough bandwidth to stream all data in real-time? Looking from a purely cyber perspective, not necessarily. Securing that connection and minimizing the attack surface is also important and needs to be balanced against what can be gained from increasing the connectivity. Thus, it is important to first determine if continuous connectivity is required. If so, then the appropriate cyber safeguards must be implemented to protect that continuous connection. The discussion of continuous versus sporadic connectivity follows.

Sporadic or Continuous Connectivity

The connectivity of the system impacts the availability of data from the system, potentially increasing latency. Systems that have a continuous connection are more likely to be able to send more up-to-date information than those that are connected only sporadically. The distinction between continuous or sporadic connectivity is not related to connections that a system is making, only with the availability of connectivity with respect to that system.

With continuous availability of a connection, on-demand data streaming via long-lived connections are effective. Information can be pushed or requested from the system on-demand or periodically without the overhead of first establishing or waiting for connectivity.

Sporadically connected systems are not as well suited for automated on-demand data transfer. On-demand data requests initiated from outside the system can't be guaranteed. To gather the same amount of data from a sporadically connected system as one with a continuous connection, the system will need a larger local data cache to store the collected data while it waits for a connection to become available.

In the case where no meaningful networked connectivity is available, data can be transferred from the system via external mass storage devices such as external hard drives, secure digital cards, compact flash, or USB flash drives.

Network Edge Nodes

As with many terms in computing, this term means different things to different people; in this context, it refers to a property in a graph or tree-like structure where we are mapping connectivity or the ability to act upon other nodes. These are of interest because they offer natural points for collecting raw data and also can offer a way to distribute the load of processing that data. They also often act as gateways from one area of the data sphere to another.

Typically, an edge node is a device connected to a network with compute capacity and placed outside a traditional local or cloud datacenter. The term can be applied to many things, from IoT sensors and managed network devices to general-purpose computers. For example, in the case of the factory use case, a jumpbox that is dual-homed on two network segments may be considered an edge node – it is a compute-capable device and can act as an edge gateway between two networks. Edge nodes can help to collect and process data to reduce the load on the network and distribute compute requirements. Examples of devices that could be components of an IIS at an edge node:

- A jump-box acting as an edge gateway allowing for remote access to systems within a network segment.
- A managed network device such as a VPN server, network gateway, firewall, or proxy server.
- An IoT sensor that can process and pre-filter data.

Type of Connection

The physical network connection influences the data throughput and the reliability of the connection. For example, it is possible for serial bus connections to be low latency and high throughput (i.e., PCI Express) or significantly slower (i.e., RS232). In both cases, however, the topology and protocols are designed for a relatively small number of co-located nodes, making scaling difficult. Wired Ethernet connections are high-throughput, scalable, and pervasive; these will often be the best option.

Some systems may have hardware debug and in-system programming (ISP) or test access port (TAP) interfaces such as Joint Test Action Group (JTAG). These connections are used to flash firmware, debug, and validate programmable devices. The firmware can contain a bootloader with provisions for reprogramming using alternate interfaces such as RS232, I2C, and SPI. With more processing power and convergence toward SoC's, this increasingly can also include higher-level interfaces such as Ethernet or WiFi for over-the-air (OTA) updates.

Ideally, wired network connections will have the lowest latency and error rate with the highest throughput, but in some cases may not be feasible to implement. Depending on the use case, several competent wireless technologies exist, such as 802.11ax (capable of over 9 Gbit/sec data transfer). Efficient low-power wireless such as ZigBee M2M is capable of 250 kbps, typically has a shorter range, and has better low-level IO than WiFi transceivers.

Once a connection becomes available, the system can establish a connection to a data collection system, send the cached data, and free up cache space. The frequency at which connectivity is available, and the performance of the connection, when available, should be known. This information will help in selecting the types of data to be recorded, how the data is being stored, and the size of the local cache.

2.3.2.1.3 Security

Security considerations will vary wildly between different types of IIS and use cases (i.e., shop floor, safety-critical aircraft systems, IT/OT networks).

2.3.2.2 Data Acquisition Sensors (DAS)

From the perspective of the CAC, if a DAS didn't see it, it didn't happen.

Ideally, the domain stakeholder defines the CSDS-related goals, selects analytical functions that support those goals, generates a list of required input data, and determines placement of data acquisition sensors to capture that specific data. Constraints such as compute capacity, storage, bandwidth, and compliance and certification will impact the availability of data and placement of sensors. For more information on the DAS concept, reference the AAF-2.

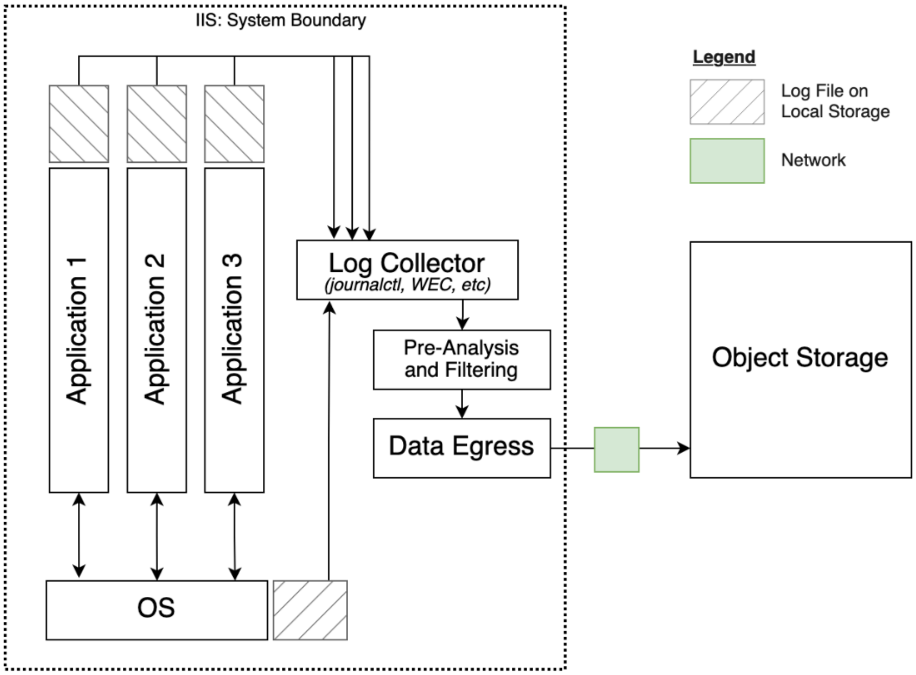

Figure 1. DAS Placed Within the IIS

2.3.2.2.1 Functions

Recommendations for sensor placement:

- Optimize the reliability, efficiency, and accuracy of the sensor. Key performance parameters must be defined to optimize the sensor, as the collected data is foundational to the CSDS AAF process.
- A particular piece of data may be available from many different locations within an environment of operation. As is practicable, select a point for data acquisition that represents the most authoritative source. Generally, this is the closest point (considering the physical distance, network topology, number of hops, etc.) as possible to where the data was created. This ensures low acquisition latency and minimizes the chance of introducing error via factors such as re-transmission, translation, or data coercion.
- Sensors should have minimal impact on the systems they are measuring. Collecting data from a system should not impact its normal operation in any meaningful way. An acceptability threshold for impact on the system must be defined for factors such as compute, storage, and system downtime.

Considerations for Industrial Internet of Things (IIoT):

- The IIoT concept describes a network of industrial tools that can share data, interact, and collaborate in a distributed way. These benefits come with the risk of potential new attack surfaces. Prior to implementing an IIoT approach for CSDS data collection, care should be taken to ensure that the increased connectivity does not undermine the effectiveness of existing security controls, such as network segmentation.

- A shop floor implementing the IIoT concept can generate an abundance of data, necessitating sophisticated data management to maximize the value of the raw data. Host systems will need to be assessed whether they can handle the heavier loads on the DAS pre-analysis and filtering functions prior to placing that system with a requirement of higher level of performance.

2.3.2.2.2 Performance

The DAS is responsible for capturing data. The sensor can be a log aggregator like Prometheus, built-in system logging like journalctl, output from components within systems like Cisco CyberVision, or even an actual physical sensor.

Depending on specific data acquisition sensor implementation, the component elements of the interconnected individual systems will have varying levels of data processing requirements. If a system is recording logs in a human-readable, string-heavy format that has few or no machine-readable fields, relevant data must be parsed, reduced, and refined from the raw data before it can be useful to analytics tools. This data-processing task can be placed at any point on, or distributed across the aggregate data pipeline from the IIS components (edge processing) to the CAC (central processing).

Examples of processing tasks that a data acquisition sensor might perform:

- Log formatting when being recorded
- Maintaining data stream from sensors to local storage
- Processing data to extract and sanitize data
- Transforming data from one format to another
- Filtering data for CSDS requirements
- Metadata collection and processing

2.3.2.2.3 Security

Security is a concern as the sensor may have low level access to a lot of information, some of which may be sensitive. Depending on the connectivity, the sensor may have to place the data in local storage for a period of time waiting to offload; this should be encrypted in the case of sensitive data. Steps should be taken to protect any data transmitted or stored by the DAS. Any collected data that is identified as critical should be encrypted at rest and in transit.

2.3.2.3 Local Storage

The local storage component in the AAF is meant to act as an ephemeral point of storage for data – it should be considered volatile storage (Figure 2).

- Data should be moved from local storage as quickly as possible to a more centralized, secured, and reliable Data-Store.
- Data should be stored in machine-readable formats (JSON, XML, CBOR, etc) wherever possible.
- Data compression can be used to reduce storage capacity requirements (gzip, 7zip), and if executed on the in-memory data stream, can also ease storage throughput requirements.

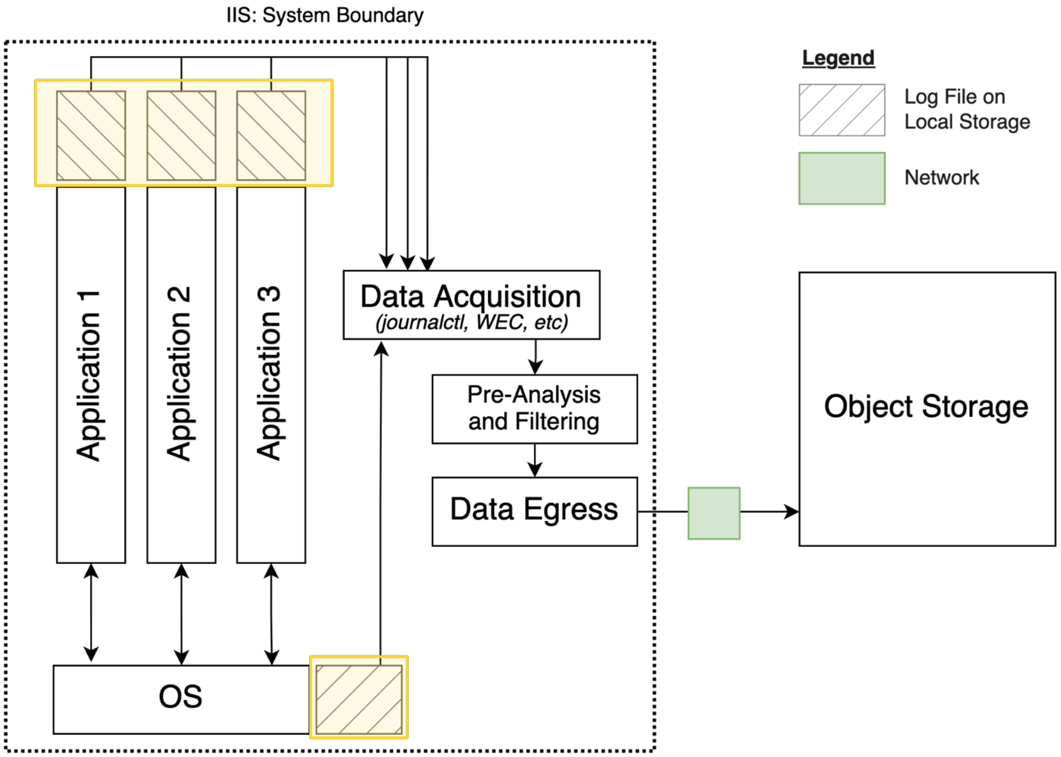

Figure 2. IIS with local storage highlighted

2.3.2.3.1 Functions

The local storage concept describes primitive storage locations that are located throughout the environment of operation. They can be embedded in an IIS (such as local disk drives and SD cards for data logging) or could be a storage-centric IIS (i.e., a storage array). While the local storage may analyze and process data with respect to its basic functions (filesystem journal, indexing, etc.), it does not analyze or process data with respect to the CSDS AAF data flow.

2.3.2.3.2 Performance

It is recommended to store the output of data acquisition sensors in a machine-readable format wherever possible. The local storage component in the AAF is meant to act as an ephemeral point of storage for data – it should be considered volatile storage in that respect. Data should be moved from local storage as quickly as possible to a more centralized, secured, and reliable Data-Store.

2.3.2.3.3 Security

Ideally, all data stored on local storage would be encrypted at rest, either in encrypted volumes or using whole-disk encryption. Realistically this is unlikely to be practicable when applying the CSDS AAF to existing systems that have strictly controlled configurations and limited resources, and so file-level encryption may be the only option available. Depending on the IIS, encryption of the data may become more critical, such as on systems that travel between sites or are exposed to the public. For more information on storage encryption, please reference NIST SP 800-111.

2.3.2.4 Data Egress Points

Data Egress Points (DEP) allow cyber-relevant data in local storage to be reliably retrieved and sent to Data-Stores (see Figure 3). They can be implemented in a multitude of ways depending on the constraints of the environment of operation.

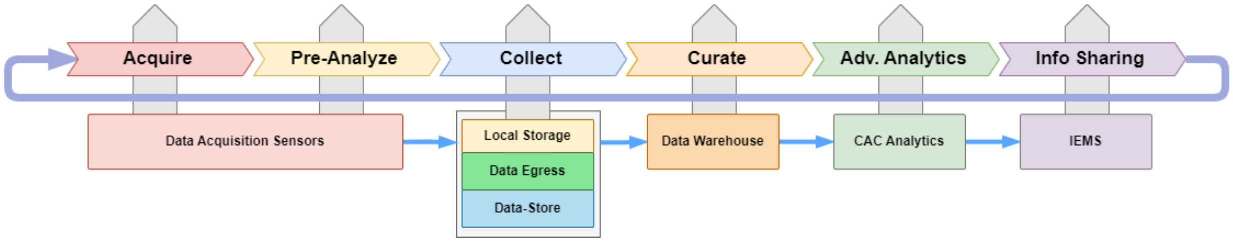

Figure 3. DEP placed against the AAF data lifecycle

2.3.2.4.1 Functions

The primary function of the Data Egress Point (DEP) is to allow cyber-relevant data in local storage (from data acquisition sensors) to be reliably retrieved and sent to Data-Stores. DEPs can be implemented in a multitude of ways depending on the constraints of the environment of operation (Figure 4). For background information on the DEP concept, please see *Part 2 CSDS AAF – Technical Definition.*.

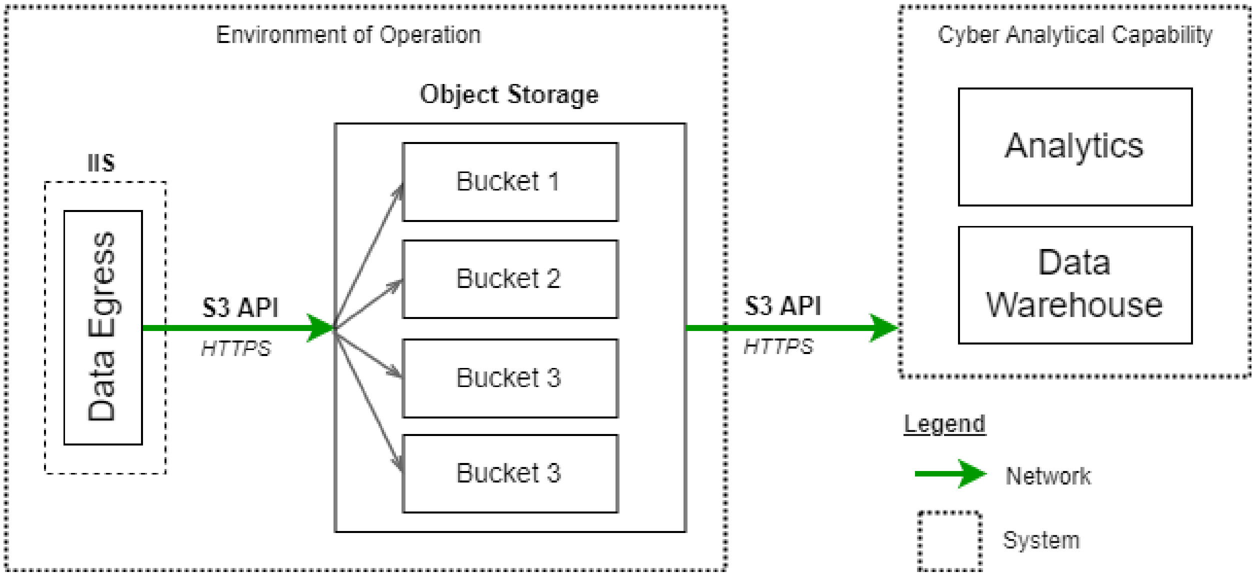

Figure 4. Data Egress from IIS to a Data-Store Implemented with S3 Object Storage

2.3.2.4.2 Performance

Metrics should be developed to measure the performance of the DEP against its primary function; by definition, these metrics should describe how well the data is being collected from local storage and sent to the Data-Store(s). Some guidance on general factors that can impact performance are listed below:

Is collection of the data from local storage manual or automatic?

- Automatic collection means that the data collected from sensors is automatically fed into the Data-Stores with no action taken by a human operator. This can happen continuously or on a set schedule and requires a data path between local storage and the Data-Store with continuous availability. Ideally, continuous data egress can yield the lowest latency and error rate when measured between when/where the data is collected and when it enters the Data-Store. Factors such as the availability of the DAS, the throughput of the connection(s) involved, and link latency may impact performance. When a continuous collection is not feasible or effective, automatic periodic collection can be implemented where data egress occurs on a fixed schedule designed around the constraints of accessing the local storage and transmitting the data to the Data-Store(s).
- Manual collection means that the data collected from the sensors is stored in local storage and then must manually be transferred by a human operator into the Data-Store. Manual collection can be done remotely, i.e., using remote data transfer with protocols such as FTP and HTTP, or via network filesystem shares such as NFS and SMB. Manual collection can also be done locally/physically using a temporary direct connection (Ethernet crossover or

temporary patch, RS232, USB host), or with removable storage (USB Flash, SD Card, external disk drive).

What is the throughput and latency of the data path?

- The throughput and latency of the data path between the local storage and the Data-Store has an impact on the availability of the data and must be balanced against the type and size of data being collected (ultimately feeding into the concept of data velocity defined in AAF-2). Sensors may store data in a raw uncompressed format. In this case, implementing continuous automatic data egress may not be possible without taking additional action to reduce the data size (i.e., compression and filtering). Different compression techniques may be used to reduce the data size of the data even further.

What is the priority of collecting the data with respect to the CSDS goals?

- Will the data help determine if there is a cyber-event pending, an attack occurring, or identify if an incident/event was caused by cyber activity? It can be difficult to tell what data could be useful in answering these questions, however, if any data is identified as relevant, its retrieval should be prioritized. Minimizing response time to threats is a primary driver in the success of implementing CSDS. When implementing collection of high-priority data, performance metrics that can be used to define minimum thresholds for data latency should be defined.
- For systems that have a large sporadic amount of data that may overload the network, a Quality of Service (QoS) profile may be utilized on a managed router or switch. Network prioritization is a common technique to ensure that packets from critical components are delivered when the network is overloaded. While employing QoS can help mitigate periodic congestion, a continuously congested network should be re-evaluated.

2.3.2.4.3 Security

Transmission of data to the data-stores should be strictly controlled. Devices and systems capable of accessing the data-store should be limited to those designated as Data Egress Points (DEPs). The aggregate DEPs are the primary data feed into the CAC, and their performance has a direct impact on the efficacy of the CSDS functions.

The DEP presents an attractive target for threat actors as a convergence of data throughout the environment of operations. DEPs will naturally be saturated with more data than other devices within an IIS, increasing the risk that a security breach of the device will yield security-critical data. DEPs also establish channels to exfiltrate data inside the EO to another location. General guidance on security and privacy controls that can be applied to DEPs can be found in NIST SP 800-53. At a minimum:

- All network communications to a DEP should use strong encryption. Selected encryption communication protocols should be standardized, such as Transport Layer Security (TLS). For guidance on selection, configuration, and use of TLS, see NIST 800-52.
- Key-based authentication should be used to control access to DEPs. All keys should be stored in a hardware security module (HSM) or appropriate secure key manager. Guidance on managing cryptographic keys can be found in NIST SP 800-57. Guidance for managing digital identity can be found in NIST SP 800-63.
- Authorization and access scopes should be restricted using least-privilege and least-functionality concepts.

For more general information on security guidelines for storage infrastructure, please see NIST SP 800-209.

2.3.2.5 Acquisition Modes (Removed)

2.3.2.6 Data Acquisition Sensor Placement (Moved)

See §2.3.2.2: Data Acquisition Sensors.

2.3.3 Stakeholder Data-Store Concept

The primary function of the Data-Store is to reliably receive and store important CSDS-related data while it awaits extraction to the CAC data pipelines and/or data warehouse. Local storage is a limited resource and is not an archive. All data must be moved to a Data-Store. There may be one or more Data-Stores within an environment of operation (Figure 5). Network segmentation, data reduction and compliance requirements, network and device performance limitations impact Data-Store locations. For more information on the Data-Store concept, please reference the AAF-2.

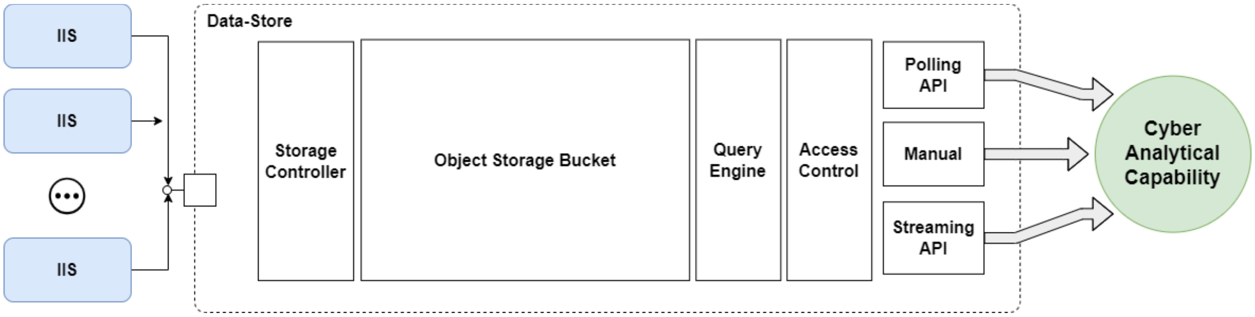

Figure 5. Diagram of an example Data-Store implementation

2.3.3.1 Functions

As local storage is both a limited resource and not specifically designed to be a reliable data archive over time, the data collected there must at some point be moved to another more specialized component in the system – the Data-Store. There may be one or more Data-Stores within an environment of operation depending on factors such as network segmentation, data reduction and compliance requirements, and network and device performance limitations. The primary function of the Data-Store is to reliably receive and store important CSDS-related data while it awaits extraction to the CAC data pipelines and/or data warehouse.

2.3.3.2 Performance

In general, data storage follows a pattern of increased cost with performance that can jump orders of magnitude between storage technologies. This relationship and the location of various storage technologies with respect to performance (color coded) is illustrated in Figure 6: Data-Store Storage Performance Considerations. Stakeholders should verify that data storage devices, such as Data-Stores, can handle the data they are required to ingest. When possible, it is recommended that actual system data is used to create a set of requirements for data storage device performance vs. generic synthetic benchmarks (Figure 6). If the Data-Store is unable to meet the desired data velocity requirements, either the Data-Store performance can be increased, or the incoming data can be further pre-filtered or reduced by the Data Acquisition Sensors or further processed by the DEPs.

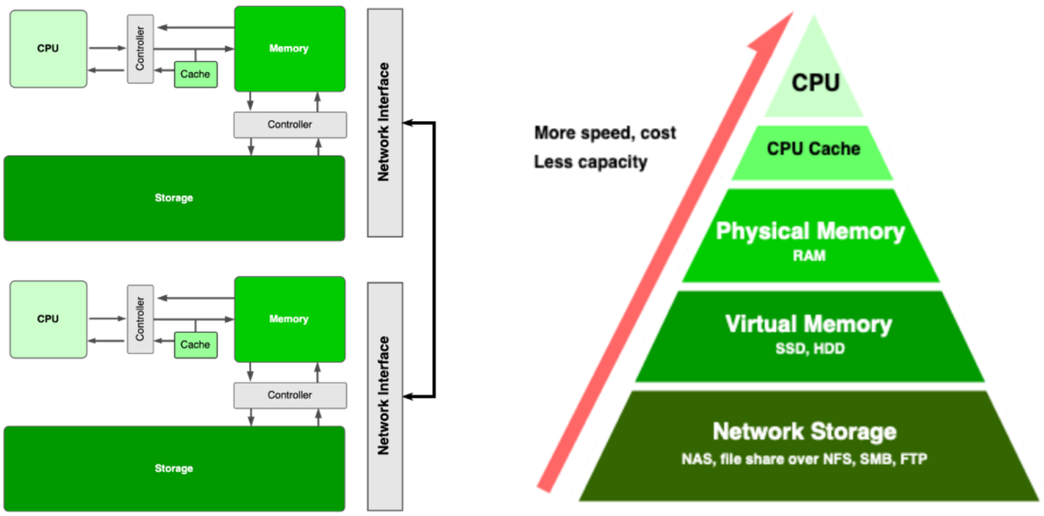

Figure 6. Data-Store Storage Performance Considerations

Each Data-Store will be required to handle the ingestion of data from one or more Data Egress Points (DEPs). There are several contributing factors in determining data velocity requirements for a Data-Store:

- The number of DEPs that are transmitting data to the Data-Store.
- The volume of data being sent from each DEP.
- The distribution of data transferred from a DEP over time. Is the data sent in fewer large packets or distributed across many smaller packets?
- The time-sensitivity of the data ingested. Is some or all data required to be processed within a certain amount of time to be useful?
- The life of the data in the Data-Store before it is extracted

Network-attached storage (NAS) or storage area network (SAN) solutions are popular and affordable data storage solutions that often implement redundant arrays of independent disks (RAID) to take advantage of the low cost of hard disk drives while parallelizing read and write operations across drives to maintain high data throughput (data striping and mirroring). An efficient and reliable middle ground would be a RAID-5 configuration, striping data with parity across a minimum of three drives.

Data throughput and redundancy can also be achieved through hybrid hardware and software-based storage solutions, such as Red Hat's Ceph Storage. This approach allows for fine-tuned control of the storage configuration to the needs of the use case at the expense of added complexity.

The choice of a data storage solution(s) should primarily be based on meeting the data velocity requirements for the Data-Store for both the present and the future.

2.3.3.3 Security

The Data-Store is an attractive target for threat actors as a centralized location of information and a nexus of communication between multiple IIS within the environment of operation. In general, the threats that must be considered are:

- Compromised or stolen credentials
- Unpatched vulnerabilities
- Scope or privilege escalation
- Ransomware
- Data loss via direct or indirect misconfiguration
- Man-in-the-middle attacks or network eavesdropping

A lapse in addressing these threats can result in:

- Data breaches, corruption, and unauthorized alteration of data
- Denial of service
- Compromised software related to data storage for the purposes of exploiting other systems on the network

For a comprehensive discussion of these threats, attack surfaces, and potential outcomes, please see *NIST SP 800-209: Security Guidelines for Storage Infrastructure, §3*.

2.3.4 Cyber Analytical Capability

Cyber Analytical Capabilities (CAC) are represented by a collection of Human Analysts using software-based toolsets to perform analytics on data to produce cyber-analytical information within a highly secured networking environment. While it is ideal for CAC to be fully contained in one physical facility with Human Analysts working from dedicated workstations connected to the local LAN, it may not be possible or feasible. Human Analysts may be located anywhere in the world and may even work remotely. For more information on the CAC concept and multi-domain environment (Figure 7), please see *Part 2*.

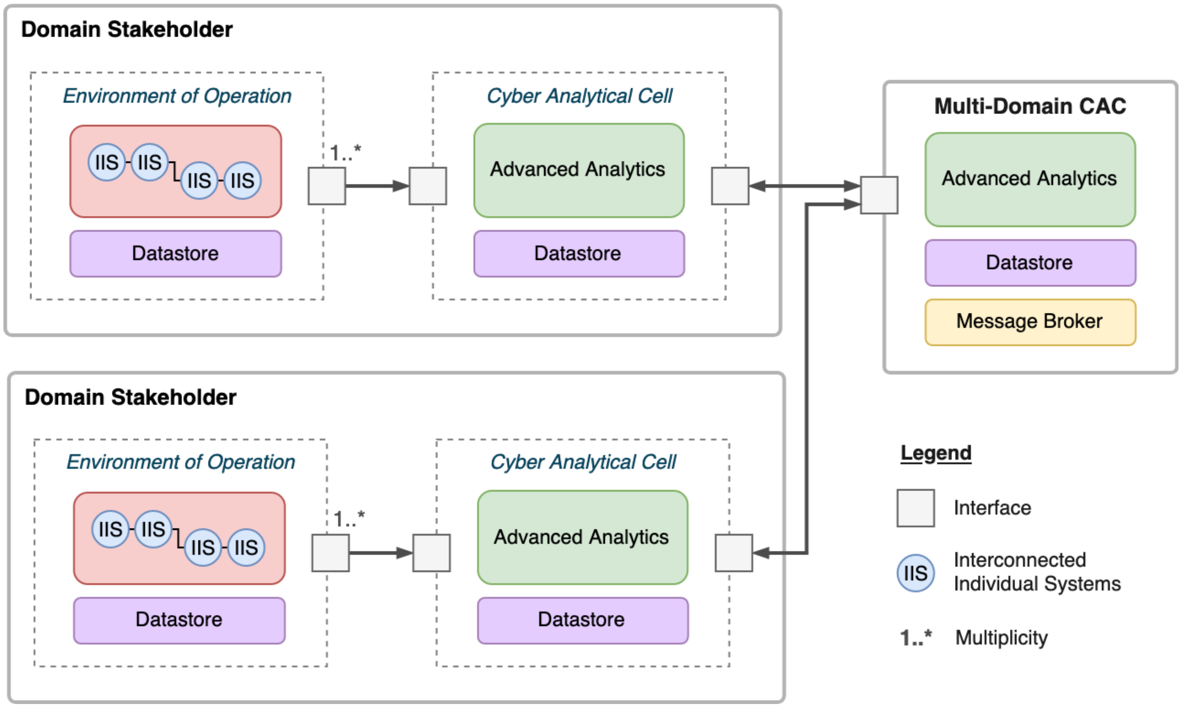

Figure 7. CAC and Multi-Domain CAC

Human Analysts in the CAC will need access to controlled software and data, such as the Data-Warehouse and Data-Store, when working remotely as well as in the office. Some security and operational challenges can be mitigated using secure thin clients to access centrally located and managed physical and virtual resources, maintaining a uniform configuration regardless of the analyst's physical location. Packaging and deploying some toolsets may require a local installation on each device used by analysts. Some toolsets will require the data they work with be stored locally; however, data retrieved from the Data-Warehouse should be managed carefully, as this can cause conflicts between different versions of the data. Recent events have necessitated a focus on remote work cybersecurity, generating a number of resources for information and best practices (i.e., Cybersecurity Risk and Mitigation Techniques During COVID-19 by Hossain, Riad, Shahriar, & Valero, 2021).

The CAC needs access to controlled software and data, such as the Data-Warehouse and Data-Store. Some security and operational challenges can be mitigated using secure thin clients:

- Access centrally located and managed physical and virtual resources.
- Maintains a uniform configuration regardless of the analyst's physical location.
- Easier to manage with central administration.

Packaging and deployment of some toolsets may require local installations and local datasets:

- Modification to local datasets should be tracked, and artifacts generated from modified datasets should be traceable back to that specific version.
- Using tools like Git LFS can help manage dataset versioning and workflow.

Access to certain toolsets or subsets of data may be limited to certain Human Analysts at a CAC. Reasons to restrict access may include:

- Access not required (least-privilege, limited scope)
- Lacking relevant training
- Compliance requirements
- Clearance level requirements

The most common method for restricting access to restrict access to networked resources has been perimeter-based and knowledge-based security, where an account and password can be used to pass through various perimeters of security. This strategy has been effective in the past, but has long been in retrograde with respect to the ever-evolving cyber-security landscape and increase in sophistication of threat actors. Ideally, a zero-trust architecture (ZTA) for shared resource access would be implemented, allowing for dynamic access control based on a wide range of variables. Please reference section 2.3.5.3 *Data Warehouse: Security,* for more information.

In a Cloud-Implemented CAC, the functions of the CAC can be implemented via a variety of cloud services that can generally be described by Platform as a Service (PaaS), Infrastructure as a Service (IaaS), or Software as a Service (SaaS). Please see NIST SP 800-145 for more information on definitions of cloud computing resources. Ideally, a cloud-based CAC would be hosted on services available from an established cloud provider such as Amazon Web Services (AWS) with GovCloud, Microsoft Azure, or Google Cloud Platform. With solutions such as AWS GovCloud, these options are certified to meet the needs of government customers that comply with the FedRAMP High baseline; the DOJ's Criminal Justice Information Systems (CJIS) Security Policy; U.S. ITAR; EAR; Department of Defense (DoD) Cloud Computing Security Requirements Guide (SRG) for Impact Levels 2, 4 and 5; FIPS 140-2; IRS-1075; and other compliance regimes.

2.3.5 Data Warehouse

Data Warehouses provide localized long-term storage for Curated Data and Shareable Artifacts for CACs of both individual stakeholders and multi-domain users to enable a long-term analysis of patterns for multiple CSDS Use Cases. For more information on the Data Warehouse concept, please see *AAF-2*.

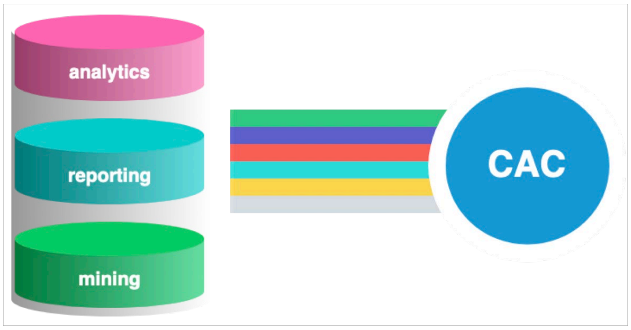

Figure 8. Three types of data stored in a Data Warehouse

When selecting or developing a data warehouse, these performance characteristics should be addressed:

- Determine necessary capacity and scaling plans.
- Ensure granular metadata and object versioning.
- Determine the average and peak number of requests to the API and any desired query processing.

- Identify the frequency and size of data transfer (ingress and egress).

The data warehouse will be a high-value target for threat actors; access should be strictly controlled, and all accounts should use multi-factor authentication (MFA), preferably possession-based biometrics, FIDO2, etc. Where possible, apply Zero Trust Architecture (ZTA) principles and control access on per-request basis with a policy engine (PE).

2.3.5.1 Functionality

The Data Warehouse functions as long-term data storage, and high storage capacity is required. The Data Warehouse should expect millions of reads and writes per day as new data is generated and old data is referenced. The data should be stored using a scalable structure that supports granular metadata. Object storage systems are recommended because they are easily scalable, support granular metadata, and are best suited for relatively static data storage. Several solutions exist for object storage, with many supporting the AWS Simple Storage Service (S3) API. Metadata is critically important to the CSDS AAF to maintain the context of the collected data. In some tasks, the meta-data may be as useful as the data itself, such as in enforcing regulatory requirements for ITAR or PII.

2.3.5.2 Performance

Performance requirements for the Data-Store are driven by factors such as number of requests to the API, size and frequency of data transfer, data structure, and query processing. When designing or selecting a Data-Store, a benchmark can be used to define the level of performance of a particular solution and used to evaluate against the expected performance metrics. The major factors contributing to the performance of the Data-Warehouse are:

How often will the Data-Warehouse be accessed?

- Generally, the speed of storage media is proportional to the cost. For example, processor cache memory is many orders of magnitude faster and more expensive than secondary storage. This can also apply to specific groups of data within the warehouse.
- The number of users accessing the Data-Warehouse and the frequency of those accesses will have an impact on the system's ability to address each request in a timely manner. The frequency of access and the size of the transaction (which is discussed later) are both elements comprising the concept of data velocity, which is defined in Part 2.

What is the average size of data in transactions to and from the Data-Warehouse?

- Larger objects being written to or read from the Data-Warehouse will naturally take longer to transmit. The average size of transactions will impact the system's performance. Transaction size and frequency are elements comprising the data velocity concept, defined in Part 2.

Other than a stronger connection, low performance can be resolved with faster reads and writes at the local level. For extremely large storage arrays, a robust software-defined storage solution such as RedHat Ceph Storage might be appropriate, allowing the Data-Warehouse to scale up to any size necessary with fast, atomic transactions in a fault-resistant array.

2.3.5.3 Security

The Data-Warehouse is a priority target for threat actors looking to exfiltrate, corrupt, or deny access to data. For example, a compromised user account (stolen credentials, SIM swap attack) could allow a threat actor to collect or corrupt data, plant ransomware, or initiate a denial-of-service attack while evading detection. Possession-based authentication methods such as local biometrics, FIDO, WebAuthn are preferred over knowledge-based auth and credentialing such as SMS one-time-passwords (OTP) and knowledge-based authentication (KBA). Due to the placement of the Data-Warehouse in a non-public zone, distributed denial-of-service (DDoS) attacks are not likely; however, in cases where an attack surface is present to allow for a large DDoS, many mitigating services are available from cloud providers such as Microsoft, Google, and Amazon, as well as content delivery networks like Cloudflare and Akamai.

While perimeter-based security, network segmentation, device and service hardening, and intrusion detection systems are each important elements of implementing defense-in-depth across multiple layers of the organization, it is recommended that organizations also apply zero-trust architecture and concepts. The Federal Aviation Administration's (FAA) 2035 Vision for Air Traffic Management integrates the ZTA; instead of defending static network-based perimeters, zero trust focuses on securing information flow between individual assets and resources. For more information on ZTA, see NIST SP 800-207.

Access scopes for the Data-Warehouse should be strictly controlled and limited using a least-privilege strategy. Access to the Data-Warehouse should be controlled via a policy engine or enforcement point capable of implementing cryptographic controls like those used by the Data Egress Points. All transactions should be recorded in an audit log. See NIST publication SP 800-209 for more information on best security practices for storage structures.

2.3.6 Information Messaging System

2.3.6.1 Functions

The PubSub messaging pattern is advantageous for systems looking to decouple services producing information from those that process the information. It also eases the definition of asynchronous communication channels. The PubSub pattern can be applied to implementing event-based messaging in myriad ways from HTTP API-based client-server polling to decentralized embedded messaging libraries like ZeroMQ. Many cloud service providers offer a PubSub-based service for asynchronous message delivery with use cases ranging from big-data processing pipelines to mobile app push notifications.

When developing the information messaging system function for the CAC, the best fit for most organizations will be to implement a message broker. As the name suggests, a message broker acts as an intermediary between publishers and subscribers of information, controlling access, managing flow, translating between formats, and generally decoupling the applications that communicate through the broker.

2.3.6.2 Performance

When implementing the information messaging system in the CAC using a message broker, the primary performance considerations are centered around scaling the system as the rate and size of messages through the broker increases.

The utilization of load balancing techniques that are commonly used for distributing HTTP API calls across servers are not drop-in solutions, since the brokers need to accomplish some PubSub-centric goals such as ensuring that messages are received by a particular consumer at least once. For example, if placing two brokers behind a load balancer, when a client connects, it will connect to one of the two brokers. It is possible (and likely) that the next client to connect to the load balancer will be assigned to a different broker than the first client. In this situation, it would not be possible for either client to send a message to the other via a topic or queue. To implement this functionality, one approach is to ensure that the brokers are aware of each other and have a method to share messages from broker to broker. This is a capability found in many widely available message brokers, such as Apache ActiveMQ, that implement high-availability modes via primary/secondary failover, store-and-forward networks, and replicated message stores. For load distribution, store-and-forward and replicated message stores are particularly useful.

When implementing the CSDS AAF, reliable and high-performance Data-Stores is a requirement for many components, such as the CAC's data warehouse and EO's Data-Store(s). The same guidance for implementing those Data-Stores (RAID, SAN, etc.) can be applied here in replicating message stores for message brokers. This allows for a set of brokers to mirror state across each instance.

A more flexible approach is to utilize built-in broker networking, such as is available in later versions of Apache ActiveMQ and Artemis, which effectively moves the job of the load-balancer inside the broker via either static lists or dynamic discovery. This can be combined with a set of rules in the broker that defines the desired topology (store-and-forward queues, mirroring, peering, etc.).

2.3.6.3 Security

An Information Messaging System should have the following characteristics:

- Authentication and authorization of each request to and from the system
- Strong encryption and signing for all messages
- Sandboxed environment for reading or parsing messages
- Anonymous messaging
    - Anonymity should be applied when a message is being sent between domains
    - Recipient should not learn who sent the message
    - Reports of messages sent should be available for administrators to search
- Enforced redaction of confidential/proprietary/sensitive information before a message is sent
    - Redaction of parts of data should not affect the original data – only the copy being sent
- Configurable restrictions on the types of data that can be shared (Data Governance)

There are many potential models for an Information Messaging System that would be appropriate. One such model is the Publisher/Subscriber model. For more information on the PubSub model, please see *Part 2 CSDS AAF – Technical Definition.*

2.4 CSDS AAF Data Architecture: The Data Perspective

2.4.1 Data Relevancy and the Data Sphere

For more information on Data Relevancy and the Data Sphere please see *Part 2 CSDS AAF – Technical Definition.*

2.4.2 CSDS AAF Data Life Cycle Data Flows

For more information on the CSDS AAF Data Life-Cycle please see *Part 2: Technical Definition.*

Data processing requirements for CSDS can be divided into two classes: system and user-oriented requirements. System-oriented data processing requirements are driven by the amount of information being processed by the system and tradeoffs between compute, storage, and bandwidth constraints. User-oriented data processing requirements are driven by user requirements such as service availability and response time, which in turn are generated or driven by business goals. The cyber-security related goals within the domain stakeholder's organization should be identified, along with how they relate to potential CSDS outputs. The overlap between the cybersecurity goals and CSDS outputs should be reflected in the objectives of the CSDS implementation within the organization. Examples of domain stakeholder cybersecurity-related objectives:

- Decrease time-to-detect
- Decrease dwell time
- Decrease recovery time
- Increase cyber-resiliency
- Reduce operational burden related to cybersecurity activities

Working from these objectives, a domain stakeholder can identify:

- The types of data that need to be collected
- The points within the environment of operation where the data can be acquired
- What type of sensors can capture the data with what level of fidelity and frequency
- Data pipelines for getting the acquired data back to the CAC for analysis
- The types of analysis that will provide actionable intelligence with respect to the objectives along with the necessary outputs of the CAC analytical functions
- New processes or process improvements needed to act on CAC outputs with respect to the objectives

2.4.2.1 Acquire

The goal of the Acquire phase is to capture data from the IIS that will be useful for CSDS efforts. The primary actor in the AAF for the acquire phase is the data acquisition sensor. When implementing the AAF, the types of data that need to be collected about the various IIS within the environment of operations should be defined as well as the locations where that data is available. Given this information, existing data telemetry and sensors can be identified, and any gaps in observability can be cataloged. For any observability gaps, existing sensors can be extended or duplicated to gain visibility to the data, or new sensors could be defined and placed. When changing existing telemetry gathering or sensing configurations, it is important to ensure that these changes can be accommodated with respect to compute capability (i.e., limited compute on embedded devices versus general-purpose computers), increased memory footprint, and storage speed and capacity requirements.

2.4.2.2 Pre-Analyze

Bandwidth for data egress from local storage is not unlimited, and some filtering and pre-analysis will be necessary to reduce and process the data. Depending on specific data acquisition sensor implementation, the component elements of the interconnected individual systems will have varying levels of system-oriented data processing requirements. If a system is recording logs in a human-readable, string-heavy format that has few or no machine-readable fields. In that case relevant data must be parsed, reduced, and refined from the raw data before it can be useful to analytics tools in the CAC. This data-processing task can be placed at any point or distributed across the aggregate data pipeline from the IIS components (edge processing) to the CAC (central processing). Examples of processing tasks in CSDS components:

- Log formatting when being recorded
- Maintaining data stream from sensors to local storage
- Processing data streams to extract and sanitize data
- Transforming data from one format to another
- Filtering data for CSDS requirements
- Maintaining and servicing data pipelines to CAC

2.4.2.3 Collect

During the Collect phase, the data may need to be restructured or partitioned to meet storage requirements and to allow for access and querying of the data. The selection of Data-Store has a significant impact on the data; how it is accessed as well as what metadata can be stored or automatically generated. For example, many object storage solutions support generational versioning of the data, which can be useful in tracking changes to data over time. A natively time-based database (TSDB) can be particularly effective for storing time-series data. Examples of generic TSDB implementations are InfluxDB and Prometheus. Operations-specific time-series databases are often used in historian functions for operational process data.

2.4.2.4 Curate

The data should be stored using a structure that is scalable and supports metadata at a granular level. Object storage systems are recommended because they meet those requirements. They are best suited for relatively static data storage.

2.4.2.5 Advanced Analytics

Notifications could be sent via email, push notifications to apps, or published as events on specific topics on a message broker. Since email is relatively less secure than a dedicated end-to-end encrypted channel, the data included in the notification should be limited in accordance with the security of the channel being used.

Notifications should not be used as a method to transfer data in to or out of a data-store or CAC. The primary purpose is notifying a set of systems or individuals of an event, not to send all the data related to an event. The role of transferring the data related to an event should be restricted to communications channels available on the data-store and data warehouse.

2.4.2.6 Information Sharing

All data passing through the AAF should be marked with the data classification associated with, whether that be PII, ITAR, EAR, or FOUO. Data might be tagged with multiple restrictions, such as both PII and ITAR, or none. All systems that use this data must be aware of these restrictions, and any outputs they provide be marked accordingly based on the data that was used. Standards documents, such as NIST 800-122 in the case of PII, can provide general guidance on handling sensitive data.

3 Application of the CSDS AAF

CSDS AAF is proposed to facilitate the development and collaboration of CSDS techniques, tools, and processes in a systematic approach to assist cybersecurity analysts in answering three (3) key questions for aviation architectures: 1) Is there a cyber-event pending? (Initial Foothold) 2) Is there an attack occurring now? (Network Propagation) 3) Was an incident/event caused by cyber activity? (Action on Objectives). These questions can be directly mapped to the Cybersecurity Unified Kill Chain model given in Figure 9. For more information, please see *Part 1 CSDS AAF Overview & Value Proposition*.

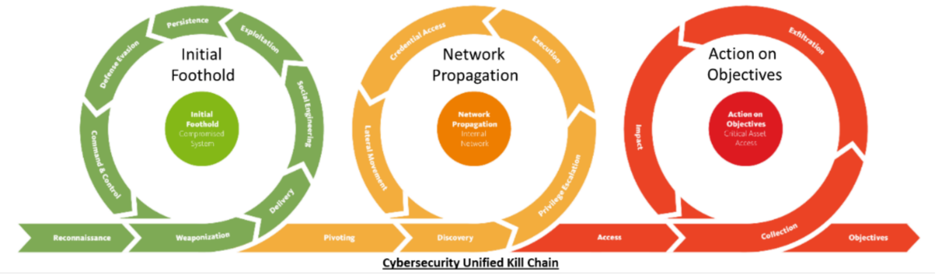

Figure 9. Cybersecurity Unified Kill Chain

One method of applying the CSDS AAF to a specific instantiation of an environment of operation (i.e. Aircraft OEM Factory) is to develop Use Cases and execute Analytical Exercises (AE) to explore and evaluate CSDS AAF application to those Use Cases via sets of scenarios.

> *CSDS AAF provides structure, key considerations, and criteria for applying CSDS methods to address the cybersecurity challenges through the development of practical aviation domain environment Use Cases. Use Cases represent the missions and goals associated with a Stakeholder's specific Environment of Operation. Use Cases will be used as the basis for implementing CSDS capabilities into the associated Environments of Operation.* [1]

3.1 CSDS AAF Analytical Exercise Concept

The CSDS AAF Analytical Exercise is a tabletop exercise where players respond to a scenario presented by facilitators outlining a hypothetical cyber-event carried out by a threat actor such as an advanced persistent threat (APT) group against an instantiation of the Stakeholder's specific Environment of Operation.[2] Facilitators present a series of facts to the players designed to appear innocuous, but in some cases representing a more serious issue. Facilitators should be knowledgeable both with the scenario and the CSDS AAF. Some of the information presented should appear contradictory, or be presented as a distraction from the real problem. The goal of the exercise is to bring the players together as a team to analyze a realistic scenario from a critical perspective, apply CSDS AAF to the scenario, and generate analysis and discussion that can be used to tailor the implementation as needed to match technical challenges and organizational goals.

[1] AAF-1 §4.4: Practical Applications of CSDS AAF.

[2] This format is strongly influenced by the CISA Cybersecurity Tabletop Exercise format.

To best execute a productive AE, facilitators and players should already be familiar with CSDS AAF and target Environment of Operation concepts and terms. Workshops and training should be provided as needed before execution of an AE.

3.1.1 Use Case Development

Use Cases can be mapped to the CSDS AAF for different Environments of Operations in the Aviation Ecosystem. Generically, a Use Case is a list of actions or event steps typically defining the interactions between an actor and a system to achieve a goal. The actor can be a human or another external system. The development process for a CSDS Use Case involves selecting and defining the Environment of Operation intended for application, identifying Threat Scenarios, and documenting the Use Case in the context of CSDS AAF. The following is a list of key potential items to include in a CSDS Use Case Document:

- Use Case ID – A globally unique ID that identifies this document
- The Use Case – A general description of the Use Case
- Description of the Environment of Operation of Interest – Refers to the Environments of Operation that belong to various Domain Stakeholders. The choice of environment, and the specific systems of focus, is typically driven by business risk considerations based on vulnerability data and threat Intel, such as perceived Primary Adversary Objective's. This should describe elements like the operational capabilities, people, processes, tools, input and outputs, and uses of the overall operational environment.
- Identification & Description of the IIS (IIS) – The various interconnected or networked systems that make up a given Environment of Operation. This will include the OT and IT interconnects, types of networks and interfaces, and types of data associated with the various systems that define the capabilities noted in the Environment of Operations described above. This should also include a systems security analysis for the identified areas of risk, considering known vulnerabilities and threats.
- CSDS Objectives – A list of CSDS Objectives that should be accomplished to a) Detect an impending Adversary Attack, b) Detect an ongoing Adversary Attack, c) provide most attack mitigation strategies/actions after an Adversary Attack.
- Threat Actors – Threat actors, their motivation, and possible attack surface.
- Threat Scenarios – A list of scenarios representing the goals of particular threat actors.
- Location of Available Relevant Data – A list of locations where available Relevant Data may be extracted from.

- Permitted Shared Artifacts – A list of permitted artifacts that can be shared with Multi-Domain CAC for the Use-Case as well as a Data Specification on the required and optional fields. This may also include reports and file formats.

The detailed steps or items that need to be considered in building out a Use Case include:

1. Select the target Environment of Operation (or sub-set thereof) for the Use Case
    a. Identify the systems configurations for the Use Case, and their associated intended uses.
    b. Identify the applicable network configurations and their operational use within the context of the Use Case.
    c. Locate and document the related IIS devices.
2. Define Threat Scenarios
    a. Identify the primary threat actor, motivation, and goal.
    b. Identify Threat Scenarios involving the threat actor
    c. Determine the risk levels of each Threat Scenario
    d. Investigate the threat actor's ongoing activities
    e. Determine ways to mitigate each Threat Scenario
3. Identify all available data for CSDS implementation and reduce to a set of relevant data.
4. Identify the cyber-relevant data from Step 3 and select data acquisition sensors:
    a. Evaluate the information type to be collected
    b. Research the sensor types that could be used to collect data
    c. Assess sensor capabilities for the collection of selected data
5. Identify the Data Extraction Process for each Data Acquisition Sensor
6. Identify or Define Data-Stores
    a. Determine the appropriate Data-Store Configuration
    b. Determine the Data-Store element locations
    c. Consider local storage security and data retention policies
    d. Consider data management strategies to organize the acquired data
7. Identify or define Cyber Analytical Capabilities (CACs)
    a. Establish plans for a Security Operations Center (SOC) or other cybersecurity facilities for each Domain Stakeholder
    b. Identify tools for extracting data from Data-Stores
    c. Identify tools for extracting data from Data Warehouses
    d. Identify tools for extracting data from Threat Intel Feeds
    e. Identify potential AI/ML algorithms for the data Pre-analysis, Curation, and Advanced Analytics phases that takes place within the CACs

8. Identify or define/determine Shareable Artifacts
    a. Determine what type of data will be considered Shareable Artifacts
    b. Ensure all Shareable Artifacts are sanitized and redacted from all confidential and sensitive data within the CACs
    c. Ensure Shareable Artifact are formatted following the data governance guidelines
9. Identify or define the Data Warehouse for CAC
    a. Consider long term storage that can process large amounts of queries a day
    b. Establish fault-tolerant data storage
    c. Store Shareable Artifacts in the Data Warehouse
10. Identify or define an Information Exchange Messaging System (IEMS) and how it will communicate with the data warehouse
11. Identify or define Multi-Domain CACs
    a. Multi-Domain Data Warehouse for long term data storage
    b. Data Governance guidelines
    c. IEMS to communicate with each Domain Stakeholder
    d. Collect Recommendations and Improvements

3.1.2 Scenario Development

A scenario must be developed prior to the AE for the selected Use Case. The scenario should be designed to execute against a specific instantiation of the Use Case in a specific Environment of Operation. An example of how to define such a scenario is outlined below:

1. Given the selected use case and specific scenario, identify a threat actor and goal.

2. Define background details on the threat actor: what tools do they use, what sectors do they usually target, what information is already known about this threat actor by security research groups, CISA, DHS, FBI, etc..

3. Define the attack that the threat actor will carry out, and detail/expand the associated cyber-kill-chain.

4. Identify challenges the attacker would encounter and develop countermeasures.

5. Fully define the initial capabilities and limitations of the malware to be used in the attack.

6. Establish the chain of events in the hypothetical scenario, assigning specific dates and times to each event. Adding detail in this step increases the fidelity of the AE.

7. Ensure the predicted impact of each event or action matches the goal defined in Step 1.

8. Define in a document or set of slides a presentation of the events established in Step 6 that is constrained to what the players could have known at each point along the event timeline. These events should be presented as a series of facts that may appear innocuous, but in some cases represent a more serious issue. Some of the information presented should appear contradictory, or be presented as a distraction from the real problem.

If player familiarity with CSDS AAF is high, the scenario may be developed specifically to exercise the organizations implementation. Otherwise, the AE can be used as a method to additionally familiarize the players with CSDS AAF, although at a minimum they should already be familiar with CSDS AAF and target Environment of Operation concepts and terms.

3.2 Analytical Exercise Example

This section presents an example of definition, execution, and analysis of a CSDS AAF Analytical Exercise. The CSDS AAF AE defined here was conducted at the FAA Florida NextGen Testbed (FTB) located at Embry-Riddle's Daytona Beach campus on February 6, 2023.

3.2.1 Use Case and Scenario Actors

The first step was to develop a high-level overview of the scenario actors developed for the AE scenario, identifying the primary actors and the goals, tools, and methods of the threat actor.

The selected use case for this analytical exercise focused on a Third-Party Vendor performing a software update in the Aircraft OEM shop floor environment. As this exercise was executed as an example, all the elements in the tabletop scenario, such as actors, processes, and tools were created to act as analogs for their real-world counterparts. These model actors were used to represent actions that might be typical in the given scenario without drawing any specific conclusions on how the real-world counterpart would act.

3.2.2 Definition of the AE Environment

The next step was to elaborate on the environment of the AE. In this scenario, the environment was an Aircraft OEM factory. Critical to this elaboration are detailed descriptions of the networks and systems that constitute the environment, to include network diagrams. In this specific scenario, much of the environment involved the operational technology (OT) systems within the Aircraft OEM. The goal is for the AE environment to replicate as realistically as possible the actual environment.

3.2.3 Attack Scenario

An attack scenario involving the scenario actors operating within the AE environment is then methodically executed. In this scenario, the threat actor executes a spear-phishing attack on a third-party vendor that the Aircraft OEM relies on. The attack scenario describes the actions and reaction by the threat actor. I

3.2.4 Scenario Execution

The scenario execution describes the actions and reactions from the perspective of the victim of the attack scenario. In this scenario, it describes the knowledge that the Aircraft OEM has at specific points during the attack such as detection and analysis of a possible event. It is key that the actors only act and react on what they would be expected to know at that point to make this scenario as possible. Based on the attack scenario, this scenario execution could cover days, weeks, or months.

3.2.5 Applying the AAF to the AE Scenario

After executing the scenario without applying AAF, the scenario is reset applying CSDS AAF. Critical to this effort is the introduction of AAF elements such as data acquisition sensor, local storage, Data-Store, data egress points, data warehouses, and CAC. These elements should be reflected in updated system and network diagrams. The purpose of employing the AAF to the scenario is describe how it can help answer the three (3) primary CSDS objectives: Is a cyber-event pending, is there an ongoing cyber-event, and what caused a cyber-event to happen?

With the introduction of these elements, the scenario is re-executed. How does the application of CSDS AAF impact the chain of events? Would the attack been detected earlier? What kinds of data would indicate an ongoing cyber-event? Would any data collected in specific cells indicate malicious activity? Questions like these and many more such questions facilitate conversation of how CSDS AAF can be employed and how potentially effective it can be. It is critical to capture the dialogue generated by these questions as they will lead to more insights and questions. In this scenario, the CSD AAF is applied to both the Aircraft OEM and third-party vendor.

4 References

AA20-352A (2021, Apr). Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations. Retrieved from https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a

Cisco Networking Knowledge Base (2019, Jan). How to configure logging in Cisco IOS. Retrieved from https://community.cisco.com/t5/networking-knowledge-base/how-to-configure-logging-in-cisco-ios/ta-p/3132434

NIST SP 800-111 (2007, Nov). Guide to Storage Encryption Technologies for End User Devices. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-111.pdf

NIST SP 800-53 (2020, Dec). Security and Privacy Controls for Information Systems and Organizations. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

NIST SP 800-52 (2019, Aug). Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

NIST SP 800-57 (2020, May). Recommendation for Key Management: Part 1 – General. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

NIST SP 800-63 (2020, Mar). Digital Identity Guidelines. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf

NIST SP 800-209 (2020, Oct). Security Guidelines for Storage Infrastructure. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-209.pdf

NIST SP 800-145 (2011, Sep). The NIST Definition of Cloud Computing. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

NIST SP 800-207 (2020, Aug). Zero Trust Architecture. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

NIST SP 800-122 (2010, Apr). Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf

NIST SP 800-82 (2015, Jun). Guide to Industrial Control Systems (ICS) Security. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

NIST IR 8183 (2020, Oct). Cybersecurity Framework Version 1.1 Manufacturing Profile. Retrieved from https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8183r1.pdf

VMWare vSphere Docs (2020, Jan). ESXi Log File Locations for vSphere 7.0. Retrieved from https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.monitoring.doc/GUID-832A2618-6B11-4A28-9672-93296DA931D0.html