Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to IoT Security Considerations</td>
<td>3</td>
</tr>
<tr>
<td>Introduction to Cisco Kinetic</td>
<td>4</td>
</tr>
<tr>
<td>Cisco Kinetic Modules</td>
<td>4</td>
</tr>
<tr>
<td>Kinetic and Cisco’s Secure Development Lifecycle</td>
<td>6</td>
</tr>
<tr>
<td>Securing Kinetic’s Cloud Platform</td>
<td>8</td>
</tr>
<tr>
<td>Cloud Access Control</td>
<td>8</td>
</tr>
<tr>
<td>API Authorization</td>
<td>8</td>
</tr>
<tr>
<td>Exposed TCP Ports</td>
<td>9</td>
</tr>
<tr>
<td>Kinetic GMM Security</td>
<td>10</td>
</tr>
<tr>
<td>GMM Security Architecture</td>
<td>10</td>
</tr>
<tr>
<td>Gateway Management Communication</td>
<td>11</td>
</tr>
<tr>
<td>Gateway Cisco Kinetic Cloud Application</td>
<td>12</td>
</tr>
<tr>
<td>Logging in the Infrastructure</td>
<td>13</td>
</tr>
<tr>
<td>Monitoring of the Infrastructure</td>
<td>13</td>
</tr>
<tr>
<td>Device Access</td>
<td>14</td>
</tr>
<tr>
<td>Gateway Access</td>
<td>14</td>
</tr>
<tr>
<td>Remote Access to Gateways</td>
<td>14</td>
</tr>
<tr>
<td>Gateway Port Requirements</td>
<td>14</td>
</tr>
<tr>
<td>GMM Resources</td>
<td>15</td>
</tr>
<tr>
<td>Kinetic DCM Security</td>
<td>16</td>
</tr>
<tr>
<td>DCM Overview</td>
<td>16</td>
</tr>
<tr>
<td>Data Security in Gateways</td>
<td>17</td>
</tr>
<tr>
<td>Organization Access</td>
<td>18</td>
</tr>
</tbody>
</table>
Data Consumers ........................................................................................................................................................... 18
  Internal AMQP Cluster (RMQ) ................................................................................................................................. 18
  External AMQP 0.9 Broker ....................................................................................................................................... 18
  IBM Watson IoT ....................................................................................................................................................... 19
  Microsoft Azure IoT Hub .......................................................................................................................................... 19
DCM Resources ............................................................................................................................................................ 19
Kinetic EFM Security ........................................................................................................................................................ 20
  EFM Security Architecture ........................................................................................................................................... 21
  Device / Endpoint Security ....................................................................................................................................... 21
  Data Transmission Security ...................................................................................................................................... 21
  DSBroker Security .................................................................................................................................................... 22
  DSLink Security ......................................................................................................................................................... 23
  Edge Node Security .................................................................................................................................................. 23
Node permissions ........................................................................................................................................................ 24
  EFM Link Quarantine ............................................................................................................................................... 24
  Permission Values .................................................................................................................................................... 24
  Permission Groups ................................................................................................................................................... 24
EFM System Administration ........................................................................................................................................ 25
  User Provisioning ..................................................................................................................................................... 25
  Password Hashing .................................................................................................................................................... 25
  User Authorization Policies ...................................................................................................................................... 26
  ParStream Authentication ....................................................................................................................................... 26
  Custom Dashboard Security ........................................................................................................................................ 26
EFM Security Best Practices ......................................................................................................................................... 28
  Best practices when deploying EFM on a node ....................................................................................................... 28
  Best practices for communication between nodes ................................................................................................. 28
EFM Resources ............................................................................................................................................................. 28
Conclusion........................................................................................................................................................................ 29
Introduction to IoT Security Considerations

Today, in a world of evolving cyber-threats, rapidly accelerating technology, and sophisticated knowledge systems, the capability of the threat actor is greatly enhanced with tools and databases that are readily available to discover and provide information about vulnerable IoT devices. The speed at which a vulnerability can be exploited is one of the most important factors to consider when defending against a potential threat of a vulnerable device or system.

Automation is one of the principle keys to building, maintaining, and scaling resilient and survivable IoT infrastructures. The commitment to automation through software engineering allows Cisco to encode the necessary security architecture and hardened configurations into a repeatable and reliable automated process at every layer of the infrastructure. Through the use of automation, Cisco can achieve a near real-time response capability. This capability enables Cisco to adapt and defend against new threats in a rapidly evolving IoT threat landscape.

As breaches become more common, more devastating, and more broadly publicized, companies are performing assessments of their industrial facilities that include not just a review of the physical infrastructure of their operation, but the security of the supporting systems, networks and the machines connected to them. Some of the key areas of concern include:

- The physical security of the end-points – machines, devices, sensors
- Determination of which protocols need to be supported and the related security configurations
- Selection of edge/ link network technologies
- System level authorization of IoT devices connecting to endpoints, gateways or servers
- Protection/securing the data streams generated from IoT devices
- Selection of the preferred back haul network, encryption parameters, and various security technologies
- Data segmentation, privacy and related multi-tenancy
- Security operations, monitoring, collection of threat intelligence related to the IoT network
- Identification of key sources and building trust and validation mechanisms
Introduction to Cisco Kinetic

Cisco Kinetic is a horizontal software platform used by Operational Technology (OT) and Information Technology (IT) professionals to build industrial and commercial Internet of Things (IoT) solutions in manufacturing, public sector smart cities, energy, oil and gas, transportation, retail, and other industries. This paper describes how the core components of the Kinetic platform are secured.

Cisco Kinetic Modules

Cisco Kinetic makes it easy to connect distributed devices (“things”) to the network, and then extract, normalize, and securely move data from those devices to distributed consuming applications and data stores. The Kinetic platform also plays a vital role in enforcing policies defined by data owners, so that they can control which data goes where, and when.

Figure 1: Cisco Kinetic Modules

Cisco Kinetic is a distributed system of software that streamlines your IoT operations by performing the following three key functions:

- **Extract**: Data is received from disparate sources (“things”), regardless of protocol.
- **Compute**: Analysis and labelling of the data enables fast decisions at the point of action, which dramatically reduces latency and makes the most efficient use of network resources.
- **Move**: The data is routed (via labelling) to the designated application(s) in real-time. The platform distributes data on-premise, through-a-cloud, between multiple clouds, and in multi-party, and multi-location situations. Data policies are used to enforce data ownership, privacy, and security.
The Cisco Kinetic platform addresses these key functions with the following three integrated modules:

- **Gateway Management Module (GMM)**—a cloud-based module used to:
  - Deploy and manage your Cisco gateways from a remote location.
  - Deploy custom applications to those devices.
  - Access devices over a secure VPN connection.

- **Data Control Module (DCM)**—a cloud-based module that receives and routes data:
  - DCM transforms and filters sensor data, then sends the results to approved destinations, according to policies set by the data owner.

- **Edge & Fog Processing (EFM)**—an on-premise module used for computation across distributed nodes of the network, from edge to destination.

These modules work together to securely handle streams of incoming data from devices, quickly process the data, and route the data to authorized applications and systems for review and consumption by authorized individuals.

**Figure 2: Cisco Kinetic Data Flow**
The next section describes Cisco’s commitment to security and the processes the Kinetic team uses to enhance its security posture to accommodate new devices and technologies as well as to defend against emerging threats. The remainder of this whitepaper details how Kinetic’s core components are hardened and used to secure customer data throughout the data lifecycle.

While detailed functional explanations of the technologies comprising Kinetics’ core components are out of scope for this document, more information can be obtained by contacting Cisco.

**Kinetic and Cisco’s Secure Development Lifecycle**

Cisco’s Secure Development Lifecycle (CSDL), is a technical development methodology built on multiple industry security standards including CIS, NIST, CERT, ITIL and ISO. The CSDL process is a 3-phase, comprehensive assessment process involving over 130 different patterns for hardening infrastructure and devices. CSDL is a repeatable and measurable process, and an annual review is required for CSDL compliance. This annual review helps to identify security gaps as cloud technology evolves and new vulnerabilities are disclosed. The continual assessment reduces risk and increases resiliency and trustworthiness of Cisco products. The combination of tools, processes, and awareness training introduced in all phases of the development lifecycle enables a secure-by-design approach, and provides a realistic and effective method for hardening Cisco products and services in order to limit their use and function to that which it was specifically designed for. Limiting the alternative use of a product or service enhances its reliability and lends to overall performance of the solution and inevitably, customer satisfaction.

The main phases implemented by CSDL are shown in Figure 4.
The CSDL process is publicly documented here.

Because Cisco Kinetic follows CSDL, security weaknesses are identified and managed through Cisco’s vulnerability and incident management processes. For example:

- Cisco continuously conducts security scans, using a variety of vulnerability assessment tools. One such example is the Center for Internet Security-Configuration Assessment Tool (CIS-CAT).

- Vulnerabilities discovered are immediately addressed and managed by the Cisco Incident Response Team (CSIRT) and the Product Security Incident Response Team (PSIRT).

Cisco Kinetic is designed to be resilient against real-world security threats often encountered in IoT projects. Of course, security of any IoT project is dependent upon many factors beyond the scope of Kinetic, such as the secure development of “things” (e.g. sensors and controllers) and how well those devices are hardened, provisioned, monitored and updated. Because security is a continual process and IoT security technologies are continuously emerging, Cisco Kinetic is frequently updated to address new risks and threats.

More Information

The Cisco Emergency Response process facilitates a quick response to attacks in progress. Cisco may work directly with the customer to develop a shared incident response plan that is customized to address specific customer risks and concerns which may minimize the effect of current and future attacks.

To learn more about the Cisco Secure Development Lifecycle (CSDL) and other Cisco security programs, go to trust.cisco.com.
Securing Kinetic’s Cloud Platform

The Cisco Kinetic Engineering Team is committed to CSDL in every phase of the development and deployment of the Kinetic cloud-based modules. The modules discussed in this section include:

- **Gateway Management Module (GMM)**—securely configure IoT gateway devices, deploy custom applications to those devices, and manage and monitor the IoT gateway devices.

- **Data Connect Module (DCM)**—securely receive data from IoT devices and route the data to the authorized destination.

The cloud infrastructure for both of these Kinetic modules run across redundant, high-performance auto-scaling nodes, and are developed using the following:

- A defense-in-depth architectural approach using Cisco’s Secure Development Lifecycle (CSDL) process for compliance and cloud certification.

- An automated development process using continuous integration (CI) pipelines.

The Kinetic DCM/GMM infrastructure scales to meet demand as devices are added to an IoT solution. The tenant solutions will run on both Amazon Web Services (AWS) and Microsoft Azure. The hardened DCM/GMM worker nodes run services as containers in highly-isolated, hardened, container-runtime environment, based on CIS Benchmarks.

The Cloud Security Alliance (CSA) publishes the Consensus Assessments Initiative Questionnaire (CAIQ), which provides a comprehensive assessment of existing security controls. Cisco Kinetic has completed the latest version of the CAIQ and provides it through the website:


Cloud Access Control

Kinetic operates on Cisco cloud infrastructure that is administered only by authorized Cisco personnel. All Administrative access to cloud infrastructure is restricted through an internal bastion host which requires Multi-Factor Authentication (MFA) and allows traffic only from whitelisted sources. Once authentication is achieved by an authorized individual, role-based access controls (RBAC) limit access of an authorized individual or service.

API Authorization

Cisco Kinetic provides APIs for customers to manage gateways, devices and data routing policies. There are APIs that are specific to GMM features and others that are specific to DCM features. However, all APIs are hosted in the same cloud platform.

- Applications made available by Kinetic can connect over TLS only.

- Connections are authenticated using API keys, which are unique to each customer.

- An API key can provide either read-write or read-only access. Each API key is delivered only once.

Roles include: GW-Mgmt (R-W), Data-Mgmt(R-W), App-Mgmt(R-W), Org-Mgmt(R-W).
Exposed TCP Ports

Kinetic’s cloud platform is actively listening for communications over the following Internet-accessible TCP ports.

Table 1: Exposed TCP Ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>GWaaS (Gateway as a Service) portal</td>
</tr>
<tr>
<td>8883</td>
<td>Secure MQTT (Message Queuing Telemetry Transport over TLS)</td>
</tr>
<tr>
<td>9123</td>
<td>TPS (Tunnel Provisioning Service) for Call Home and provisioning of new gateways (temporary)</td>
</tr>
<tr>
<td>5671</td>
<td>Secure RMQ (Rabbit Message Queue over TLS)</td>
</tr>
</tbody>
</table>
Kinetic GMM Security

The Cisco Kinetic Gateway Management Module (GMM) is a scalable cloud-hosted application used to provision, manage, and monitor Cisco industrial IoT gateways (IR809s and IR829s). It is designed with security in mind, leveraging standard security protocols such as IPSEC and TLS, as well as the Cisco router platforms’ security architecture. Kinetic GMM is cloud native, multi-tenant, and Cisco SBP (Service Billing Platform) enabled.

GMM is used to bring new gateways online in minutes and easily manage them from a remote location. GMM streamlines provisioning and provides ongoing visibility and control of gateways from a desktop browser.

Figure 5: Cisco Kinetic GMM

GMM Security Architecture

GMM provides secure provisioning and management of Cisco IR 8x9 gateways over an IPSec tunnel. Cisco IR 8x9 gateways can be deployed with zero touch provisioning using a secure web application. Gateways are quickly configured using templates, and devices can be securely and remotely accessed using the Cisco AnyConnect secure VPN. Cisco IOx fog applications installed on the gateways can also be securely and remotely managed using the Cisco Kinetic Cloud Application.
Gateway Management Communication

Communication between the Kinetic gateways and the Cisco Kinetic Cloud Application is secured using open standards, as shown in Figure 7: GMM-Cloud Communication.
1. A gateway that is provisioned to work with Cisco Kinetic calls the "home" URL over HTTPS to register with Kinetic. Mutual authentication is performed using a Secure Unique Device Identifier (SUDI) certificate provided by the gateway and a Secure Sockets Layer (SSL) certificate provided by Kinetic.

2. When the gateway is successfully registered, Kinetic responds to the HTTPS request with the gateway configuration that is required to establish an Internet Protocol security (IPSec) tunnel with Kinetic.

3. The gateway establishes an IPSec tunnel (using IKEV2) with Kinetic. Note: the tunnel ends on a Cisco Cloud Services Router (CSR) running in the Kinetic cloud.

4. After the tunnel is established, all traffic required to manage the gateway travels through the tunnel. The traffic uses HTTPS or SNMP over IPSec. No customer data is sent for GMM gateway management.

**Gateway Cisco Kinetic Cloud Application**

Kinetic GMM provides a secured web-based application for managing and monitoring gateways, known as the Cisco Kinetic Cloud Application. This application resides in Cisco’s cloud infrastructure. Access to the Cloud Application is controlled using role-based authentication and group authorization schemes leveraged through GMM. All traffic, regardless of zone, flows securely over encrypted protocols.

In addition, all gateway management data, details, and configurations are restricted to a logical organization and cannot be shared with other organizations. Role-based access, such as admin and operator roles, manages and restricts access within the organization. Numerous policies control the level of access for each component or operator, for example several policies governing password complexity are applied in the infrastructure at discrete layers including the cloud service provider, system and application layers. Administrative access requires two or three factors in all cases.
Figure 8: Cisco Kinetic Cloud Application

Logging in the Infrastructure

All system and application logs are sent to a centralized logging system that operates as part of the infrastructure. For example, all authentication and authorization events are logged and metrics for these specific events are sent to the metrics backend for situational awareness and longer-term pattern tracking and visibility. The logging cluster is built through automation and deployed in HA configuration. The log data is backed up daily, archived onto encrypted media, and is searchable online (internal to Cisco Operations only) for 90 days. Archived data is retained for 2 years and can be available for investigation within a few hours.

The cloud service provider logs are securely relayed to Cisco InfoSec CSIRT. Within CSIRT the logs are analyzed by several security analysis platforms. Patterns and trends are identified and proactively investigated by qualified Security Investigators.

Monitoring of the Infrastructure

Kinetic employs a multi-layer monitoring strategy, which provides a very detailed level of awareness for the on-call team as well as rapid root-cause identification capabilities. The monitoring systems are hardened and built through automation and can be scaled to meet the demand of scaling infrastructure. Emitters on the systems send metrics to a centralized metrics cluster. Numerous aspects of the public cloud configuration are continuously scanned and validated through scheduled automation to ensure no unnoticed exposure or undue configuration change has occurred. Vulnerability scanners run authenticated and unauthenticated scans daily to discover any system or package vulnerabilities, a
comprehensive web application scanner runs on a weekly basis against public and private service endpoints. Notifications for infrastructure resources and services are sent to a reliable notification backend that immediately notifies the on-call team. Discovered vulnerabilities are automatically submitted to a work-tracking application and queued for due action based on severity. For example, high-severity vulnerabilities are remediated within 24 hours.

Device Access

A MQTT client runs on every gateway (e.g. IR809). The MQTT client is assigned a shared secret key that is generated by the Cisco Kinetic Cloud Application. The shared secret is unique for each gateway. Every time a gateway connects to Kinetic the connection is authenticated by this shared key and encrypted using TLS or secure web socket.

Gateway Access

Remote Access to Gateways

Admin users can use the Cisco AnyConnect VPN Client to remotely manage and interact with devices behind a Kinetic gateway. Admin users can only access devices attached to gateways within their organization.

For example, an elevator technician can create a virtual private network (VPN) between his PC and an IR809 connected to an elevator in another city. He can then use a diagnostics application on his PC to troubleshoot an issue with the elevator, determine a solution, and dispatch a repair technician with the right parts for that issue.

IP devices that use DHCP and are directly connected to the gateway can be accessed over the AnyConnect VPN connection. Devices with a static IP address require a custom gateway configuration.

Gateway Port Requirements

Outbound communication from each gateway must be permitted on the following network ports to allow the gateways to communicate with Cisco Kinetic:

Table 2: Gateway Port Requirements

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>UDP</td>
<td>Domain Name System (DNS)</td>
</tr>
<tr>
<td>123</td>
<td>UDP</td>
<td>Network Time Protocol (NTP)</td>
</tr>
<tr>
<td>500</td>
<td>UDP</td>
<td>Bidirectional access is required for the Internet Security Association and Key Management Protocol (ISAKMP) / Internet Key Exchange (IKE)</td>
</tr>
<tr>
<td>4500</td>
<td>UDP</td>
<td>Bidirectional access is required for IPSec NAT Traversal</td>
</tr>
<tr>
<td>9123</td>
<td>HTTPS/TCP</td>
<td>Call-home registration to the Tunnel Provisioning Server</td>
</tr>
</tbody>
</table>
GMM Resources

Click the following links for more information about Cisco Kinetic GMM:

- Gateway deployment requirements
- Remote management requirements
- GMM Overview
Kinetic DCM Security

Kinetic DCM ensures the security of data that is extracted, processed, and moved using Cisco Kinetic by building on the Kinetic cloud and GMM security infrastructure.

DCM Overview

The Cisco Kinetic Data Control Module (DCM) is used to securely access data sent from devices through the Kinetic IoT gateways and the EFM module(s). DCM then securely transmits that data to authorized systems (known as data subscribers, data consumers, or data destinations). Data destination options include an internal AMQP Cluster (RMQ), external AMQP brokers, IBM Watson IoT and Azure IoT Hub.

Figure 9: DCM Data Destinations

Note: Cisco Kinetic DCM does not store data. Customers can choose to direct their data to a storage solution, but any data not sent to a data destination is discarded.

When Kinetic DCM delivers data from the gateways to a data destination, DCM only accepts connections over Transport Layer Security (TLS) and strictly validates all certificates. Only TLS 1.1 and 1.2 with strong ciphers (AES128 and AES256) are supported, with future directions towards TLS 1.3.

Figure 10 shows the secure data connections used for management applications and data flows.
Data Security in Gateways

IoT Gateways collect data from connected edge devices, host the edge and fog software used to perform computations on that data, and route the data based on configured data policies. For security hardening purposes, no inbound ports are open on Kinetic gateways, including no management ports. The only mechanism for outbound communication from Kinetic gateways is through establishment of a secure mTLS tunnel with the Kinetic cloud. Only the Kinetic cloud can exchange data with the gateway, and only through this secure tunnel.

Kinetic users can run software inside the gateway that accesses locally connected devices through any of the Ethernet or Serial (RS-232/RS-485) interfaces available on the Cisco IoT Gateway.

Applications installed on the gateways always communicate with the cloud using MQTT over TLS, or MQTT over a secure web socket. Kinetic DCM does not support MQTT over TCP or unsecured web sockets. This secure connection is authenticated using a shared secret, which is generated by Kinetic and is unique for each gateway.

The following secure ports are used:
### Table 3: Secure Ports Used For Gateway Communication

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>TCP</td>
<td>Hypertext Transfer Protocol (HTTPS) over TLS/SSL Used for Web Sockets and HTTPS APIs</td>
</tr>
<tr>
<td>8883</td>
<td>TCP</td>
<td>Secure MQTT (MQTT over TLS) for the data pipeline. Required for Cisco Kinetic DCM when publishing to the Kinetic cloud.</td>
</tr>
</tbody>
</table>

### Organization Access

Each gateway, and the data from the gateway, is also assigned to a specific organization within the Kinetic Cloud Platform. The device and its data cannot be accessed by users of other organizations.

### Data Consumers

The following sections describe security aspects specific to each of DCM’s supported data consumers. For example:

- Customers can configure their applications to consume data from an AMQP Broker. This broker can be hosted inside the customers’ data center / cloud.
- Cisco Kinetic hosts an internal AMQP cluster (the cluster consists of several brokers) where the customer applications can subscribe directly to the data.
- DCM can send data directly into IBM Watson IoT Bridge and Microsoft Azure IoT Hub.

#### Internal AMQP Cluster (RMQ)

Customer applications that access data from the internal AMQP Cluster implement the following security measures:

- A secure connection to the Kinetic Cloud Platform is created over port 5671 (AMQP/TLS).
- The connection is authenticated using API keys with the data management role enabled.

Kinetic creates a pair of exchanges (observation and command) for every organization. Access to these exchanges requires the API Key for the same organization. RMQ (RabbitMQ) and AMQP concepts like exchanges are explained here: [https://www.rabbitmq.com/tutorials/amqp-concepts.html](https://www.rabbitmq.com/tutorials/amqp-concepts.html)

#### External AMQP 0.9 Broker

The following security measures are used to send data to an external Advanced Message Queuing Protocol (AMQP) host:

1. The Kinetic Cloud Platform establishes a secure connection to the external AMQP broker over port 5671 (AMQP/TLS).
2. Kinetic validates certificates of the AMQP brokers to which it connects.

IBM Watson IoT

Kinetic inter-operates with IBM Watson IoT using the following security measures:

1. A secure MQTT connection to the Kinetic Cloud Platform is created over TLS.
2. Kinetic validates the certificates of the MQTT brokers to which it connects.
3. Kinetic authenticates using the API keys for the IBM IoT account (not the Kinetic API Keys). External keys are secured as described in the CAIQ for Kinetic.

Microsoft Azure IoT Hub

Kinetic inter-operates with Microsoft Azure IoT Hub using the following security measures:

1. A secure MQTT connection to the Kinetic Cloud Platform is created over TLS.
2. Kinetic validates the certificates of the MQTT brokers to which it connects.
3. Kinetic authenticates using the API keys for the Microsoft Azure IoT account (not the Kinetic API Keys). External keys are secured as described in the CAIQ for Kinetic.

DCM Resources

See the DCM Overview for more information.
Kinetic EFM Security

The Cisco Kinetic Edge & Fog Processing Module (EFM) creates a data fabric that enables distributed processing of data from IoT devices. EFM is based on the Distributed Services Architecture (DSA). This architecture consists of Nodes (a software resource, device interface, storage, or logic), Distributed Service Links (DSLinks) that connect a Node to a Broker using the NodeAPI protocol, and Distributed Services Brokers (DSBrokers) that provide the communication fabric that interconnects all of the Nodes.

For example, Device DSLinks capture the telemetry (IoT data) from IoT devices, DSLinks transform the data into a format that can be consumed by applications or data storage services, and all are connected to the system with DSLinks. The DSBrokers transport the data using a fabric of IoT message brokers to various destinations.

More information on DSA can be found here: http://iot-dsa.org/get-started/how-dsa-works.

EFM includes a system DSLink, which is used to extract system data (including CPU and memory utilization, system time etc.). Additional DSLinks can be installed to extract data from devices and/or their controllers.

EFM also enables local storage of the data at the Edge & Fog layers using a build-in historian database.

Security must be considered at each step of the data path, as data is collected, processed, and moved from end devices to the Edge & Fog layer of the network and then to applications consuming it. Applications that consume the data can be local, in the data center, or in a cloud accessed via the DCM module. To secure this data, EFM uses a layered approach:

1. **Physical security**—Physical security of the platform where EFM is installed ensures that computing nodes cannot be altered without consent. Physical security measures authorize, control, audit, and monitor physical access to the infrastructure hardware, such as computing devices, power supplies, backup media, and network communication equipment.

2. **Host operating system security**—EFM can be installed on multiple operating systems, including Linux, Microsoft Windows, and Cisco IOx. Each OS or virtual machine environment must employ the security best practices and policies for that platform to ensure proper user authentication and authorization for administration, patch management, and storage encryption. Because EFM files are accessible to the users who install and operate EFM service(s), proper file system security must be applied to limit unnecessary file sharing.

3. **Data transmission security**—EFM supports secure encrypted communication between brokers to prevent data and control traffic from being intercepted.

4. **EFM user and node management**—EFM organizes the data hierarchically. Node management allows the administrator to assign list, read, write, and configure permissions for nodes in that hierarchy and their children.
EFM Security Architecture

IoT data needs to be secured both in transit and at rest. Data must be secured end to end, from the device or thing that generates the data to the application that consumes it, including the journey along the data path.

Figure 11 depicts the journey of IoT data in a typical Kinetic EFM-based solution and the type of security that must be applied at various stages.

Figure 11: EFM Security Architecture

The following sections define the stages and their roles in providing secure communication.

Device / Endpoint Security

An “endpoint” is the source of IoT data. Since the EFM software cannot control the endpoint device functionality, securing the endpoint is primarily achieved through means recommended by the device manufacturer. Physical and cyber security best practices should be employed including:

- Placing it in a secured and managed location
- Restricting access to the device to prevent any tampering
- Monitoring and auditing physical access to the location where the endpoint is hosted

Data Transmission Security

The Kinetic EFM system uses a highly reliable message passing system, based on IoT message brokers that establish multi-hop communications between brokers. This system sends all data through TLS connections over the network to prevent traffic monitoring. The EFM system administrator application is protected by the HTTPS protocol.
Figure 12: EFM Transport Security

EFM simultaneously works with various southbound protocols via DSLinks (Distributed System Links), a part of the Distributed Services Architecture (DSA). Each southbound connection, as shown by (1) in Figure 12, has different security mechanisms as defined by their protocols.

Depending on the IoT use-case, Cisco Kinetic EFM may also integrate with northbound applications and databases at the fog or data center layer, as demonstrated in (3), (4), and (5) in Figure 12. This communication is secured using authentication and TLS. For example, (4) JDBC over TLS is used for database integration where the Java Database Connectivity (JDBC) API is supported.

DSBroker Security

Communication between nodes is performed using NodeAPI, a stateful and lightweight streaming Remote Procedure Call (RPC) protocol. While the NodeAPI protocol can be used over different transport layers such as HTTP(s), WebSockets, sockets, or other message-passing protocols, (github) broker-to-broker communication in the EFM implementation of NodeAPI only uses WebSockets with TLS, known as WebSockets Secure. The WebSockets Secure handshake is based on ECDH with the NIST recommended curve P-256, also known as secp256r1. The key derivation function is SHA-256. For the key-wrapping algorithm, AES-256 is used, while AES-256-CBC is used as the content encryption algorithm. Both the WebSocket client and server must send a message to each other at least once every 60 seconds. If no message is received in 60 seconds, the WebSocket is considered disconnected and a re-connection is necessary. Table 4 explains the communication channels shown in Figure 12 and summarize the typical transport security mechanisms for communication through the EFM platform.
## Table 4: EFM Secure Communication Channels

<table>
<thead>
<tr>
<th>Number</th>
<th>Communication Endpoints</th>
<th>Encryption Mechanism</th>
<th>Authentication Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IoT Gateway – EFM Edge Node</td>
<td>TLS or HTTPS</td>
<td>Password or Token based</td>
</tr>
<tr>
<td>2</td>
<td>Between EFM Brokers</td>
<td>Web-socket over TLS</td>
<td>Token based</td>
</tr>
<tr>
<td>3</td>
<td>EFM Broker – ParStream (Historian)</td>
<td>VPN Tunnel</td>
<td>Password based</td>
</tr>
<tr>
<td>4</td>
<td>EFM Broker – Database</td>
<td>JDBC over TLS</td>
<td>Password based</td>
</tr>
<tr>
<td>5</td>
<td>EFM Broker – Northbound Application</td>
<td>TLS or HTTPS or VPN Tunnel</td>
<td>Password or Token based</td>
</tr>
</tbody>
</table>

### DSLink Security

EFM establishes southbound communications to endpoints via DSLinks. If the EFM Edge Node is configured to not allow unknown DSLinks, then an API token is required to establish a new WebSockets. A secure handshake occurs between the DSLink and the DSBroker running on that Edge Node. A token is not required for DSLinks that are managed by an individual broker’s DSLink manager.

- If a DSLink is installed using the broker’s `/sys/links` path or using the EFM System Administrator, that DSLink uses an automatically generated token.

The token for link authentication uses the following parameters:

- **Time Range**: A date and time string that indicates when this token is valid.
- **Count**: An integer that limits how many times this token can be used.
- **Managed**: A Boolean value. When TRUE, the token is managed, meaning that token expiration or removal causes all EFM Links connected via this token to be disconnected and removed from the EFM broker.

### Edge Node Security

A “protocol bridge” DSLink performs a specific micro service. Recall that the DSLink resides on the Edge Node that is connected to endpoint devices. The DSLink could be a bi-directional bridge for data transmission between an industrial automation control protocol (natively supported by an endpoint such as a Programmable Logic Controller) and the messaging broker’s native communications. From the perspective of the Edge Node:

- Northbound communication is always to the EFM message broker and uses the NodeAPI communications protocol and node authentication.
Southbound communication can be performed over the network or using serial interfaces (using a large variety of protocols). Optimally, the DSLink should be as close to the data source as possible since legacy protocols provide little or no authentication over serial interfaces. Networking protocols require authentication and can support secure transmission.

Node permissions

EFM Link Quarantine

Kinetic EFM broker places unknown Links in quarantine. The Link cannot be used with the EFM system until the administrator explicitly authorizes it.

If the broker is configured in Quarantine mode, all new nodes can be registered with a broker, but requires administrator authorization before any data is exchanged.

Permission Values

Individual node permissions can optionally be used to restrict the capabilities of individual DSLinks.

Permissions levels define the actions that can be performed by a user or DSLink, and the functionality of additional tools such as the Dataflow editor. For example, a user with “list” permission can only view the path tree node names. Values are not displayed or modified, and new nodes cannot be configured when assigned the “list” permission.

Table 5: EFM Permission Levels

<table>
<thead>
<tr>
<th>Permission Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No access privileges are provided</td>
</tr>
<tr>
<td>list</td>
<td>Attributes can be read, but not the values</td>
</tr>
<tr>
<td>read</td>
<td>Values and attributes can also be read</td>
</tr>
<tr>
<td>write</td>
<td>Values and attributes can also be written</td>
</tr>
<tr>
<td>config</td>
<td>(Maximum permission level) Configuration is also allowed</td>
</tr>
</tbody>
</table>

Permission Groups

Each DSLink or user can belong to one or more permission groups. Valid permission groups are:

- Username
- Custom user group
- “default”
Permission groups are created:

- Automatically when a username is created
- Manually when a custom user group name is defined

A username or DSLink that’s been assigned the global permission value “config” can assign permission groups to any user or DSLink. Custom user groups can be defined at startup time or using the EFM System Administrator (go to Permissions > Groups). If a token has a default permission group defined, DSLinks connected via that token have this permission group by default.

When connecting a broker to another broker, a permission group can be defined to limit the upstream broker’s access back to the downstream broker. For example, if the upstream broker is not allowed to write to the downstream broker, a permission group “read” with read only privileges can be created and used when configuring the new upstream connection parameters.

EFM System Administration

User Provisioning

EFM allows user management through the System Administration application, as shown in Figure 13.

Figure 13: EFM User Management

Password Hashing

Passwords created as part of user provisioning are stored inside the broker server folder. In EFM, PBKDF2 applies a pseudorandom function, such as hash-based message authentication code (HMAC), to the password along with a salt.
value and repeats the process many times to produce a derived key, which can then be used as a cryptographic key in subsequent operations. The following parameters are used with PBKDF2 key derivation function on EFM:

- 1000 iterations to generate the hash key and key-length of 32 bytes as default values. In the newer releases of EFM, the default is changed to 10,000 iterations.
- The HMAC algorithm is SHA-256.
- A salt of 32 bytes is used.

User Authorization Policies

A user can be assigned different levels of authorization by using Permission groups. A user can be assigned to a specific group with different authorization levels using the Permissions tab in the System Administrator application.

ParStream Authentication

The ParStream database does not have its own identity store for authentication. It uses Pluggable Authentication Modules (PAM). It is possible to use a wide range of authentication modules (e.g. LDAP). All rules regarding hashing and salting apply as configured for the Linux server. Note: This creates a strong dependency of security hardening the Linux server in order to secure the ParStream database.

Custom Dashboard Security

EFM allows custom dashboards to be created and connected to a Fog Node through the northbound API. These dashboards typically establish a connection to the Fog Node using HTTPS or TLS. There are multiple authentication options when establishing these connections:
Table 6: Authentication Options

<table>
<thead>
<tr>
<th>Sr</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>file</td>
<td>Default login mechanism. With this option, the dashboard authenticates users against the ‘users.json’ file, which was created using user management through the System Administration application. Super user privileges can be granted using this option.</td>
</tr>
<tr>
<td>2</td>
<td>LDAP</td>
<td>This option is for Active Directory (AD) integration with LDAP as authentication protocol. The LDAP AD server details are configured in the ‘server.json’ file. This also supports fallback to secondary server or local file users using following options: ‘ldap-with-fallback’ and ‘ldap-with-fallback-and-file’.</td>
</tr>
<tr>
<td>3</td>
<td>proxy</td>
<td>This option is for Active Directory (AD) integration with Kerberos as authentication protocol in an Integrated Windows Authentication (IWA) environment. There is a reverse proxy required between EFM Dashboard and AD server for this option.</td>
</tr>
<tr>
<td>4</td>
<td>open</td>
<td>This authentication type will allow any username/password combination to login. Superuser privileges are never granted with this option.</td>
</tr>
</tbody>
</table>
| 5  | composite | This option enables a sequence of the preceding authentication methods. Examples include:  
  - composite[ldap,file]  
  - composite[file,open] |

Figure 15: EFM Dashboard Security
EFM Security Best Practices

Cisco Kinetic EFM employs multiple security features to increase data security and privacy of data sent from the sensor to the destination.

Due to the flexibility of the EFM module, best practices for EFM installation and node communication have been established.

Best practices when deploying EFM on a node

- Harden the host-operating environment as much as possible. Update the OS and configure user access to prevent unauthorized access.
- Apply file system encryption when available.
- Enable firewalls or networking filtering on routers or switches running IOx to limit unnecessary inbound and outbound connections.
- Install the EFM and components to a non-root (on Linux) user.
- Install the EFM message broker in the secure mode, which restricts unencrypted incoming connections to the message broker (HTTP connections are not allowed).
- Enable node quarantine so unauthorized node connections to the message broker are not allowed.

Best practices for communication between nodes

- Use encrypted broker to broker communications over WebSockets (HTTPS connections).
- Apply networking filtering between endpoints to only allow:
  - Northbound communications between message brokers and micro services.
  - Southbound communications between the message broker or micro services and sensors, actuators or controllers.

EFM Resources

See the EFM Overview for more information.
Conclusion

Digitization and IoT address complex business problems and create real value in an environment of increasing regulation, immense scale, and unprecedented combinations of IT, OT, and multi-vendor complexity. The Cisco Kinetic platform provides a solid foundation to securely manage very large-scale IoT systems and their data.

The Cisco Kinetic platform enables users to build secure end-to-end IoT solutions in a very flexible manner. Good judgement and operational security expertise are paramount when implementing, operating, and maintaining Kinetic deployments in order to increase the cyber resilience of Kinetic-enabled IoT solutions.

As evident throughout this document, different security precautions are required between the cloud and the on-premise domains, dealing with both the data path and the control plane, and between trusted environments protected by physical security and unregulated environments that may span geographies and/or corporate boundaries. The scope and span of security protections across the different Kinetic modules is a direct response to the many issues present in systems of this scale and diversity. Cisco has leveraged its immense experience and deep knowledge of networking and security to enhance the cyber resilience of the Kinetic platform, so that our customers can realize the potential of IoT and digitization while minimizing associated risks.

Note: For more general information and FAQs related to Cisco’s Security Compliance Program and Cisco’s GDPR readiness, please visit the Cisco Trust Center at trust.cisco.com.