Eagle.io Business Security

An Eagle.io Whitepaper

Limitations

This report has been prepared by Argos.io pty ltd t/as eagle.io (eagle.io) in accordance with our standard terms and conditions.

We may at our discretion, but without being under any obligation to do so, update, amend or supplement this report.

This report must not be copied, reproduced, distributed, or used in whole or in part for any purpose without the written permission of eagle.io.

Version register

Version	Status	Date	Author	Reviewer	Change from Previous Version
2021.01	First release		RA	BS/DJ/JW/JM	-
2021.02	Released	24-Sep-2021	RA	JW	WAF integrated
2024.01	Released	8-Apr-2024	RA	BS	ISO program migration

Contents

1 Introduction

Eagle.io provides a software as a service product that is accessed via a web browser or programmatically through an API. The platform contains a significant amount of information. We've designed an ISO27001 compliant security framework to ensure that eagle.io applies a consistent, risk-based approach to the implementation of information security to maintain confidentiality, integrity and availability of the platform.

In this whitepaper, we'll detail the back-end policies, as well as options available that make eagle.io a secure tool for clients.

2 System Architecture

Our interfaces are driven by a backend infrastructure optimized to ensure a fast and reliable service. We are continually evolving our product and architecture to improve reliability and speed. In this section, we'll explain how data is transferred, stored, and processed securely.

Eagle.io's system infrastructure consists of the following components in Figure 1.

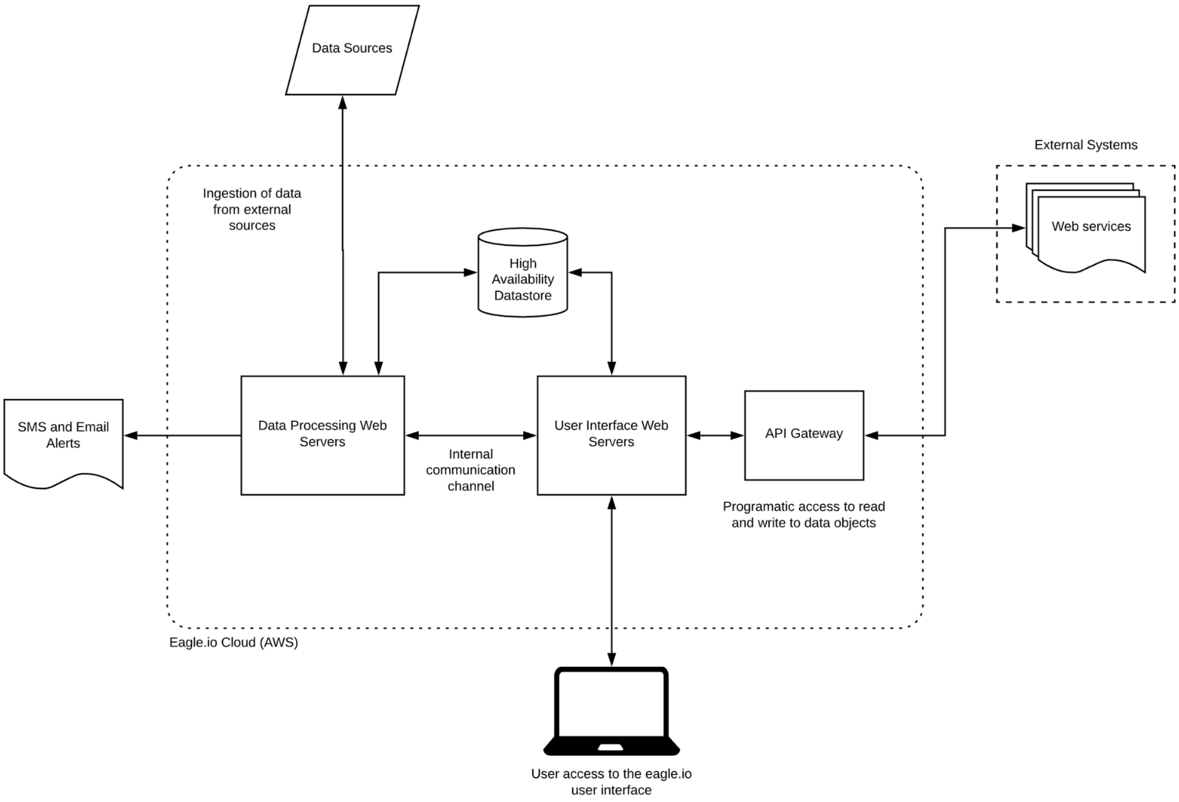
Data Sources
External Systems
Web services
Ingestion of data from external sources
High Availability Datastore
SMS and Email Alerts
Data Processing Web Servers
Internal communication channel
User Interface Web Servers
API Gateway
Programatic access to read and write to data objects
Eagle.io Cloud (AWS)
User access to the eagle.io user interface

Figure 1 - High-level architecture

Data processing web servers

This component comprises of 5 independent services:

1. **Portserver** is an internet-facing service that receives all the incoming connections from remote data sources via various protocols. Data is stored and queued to allow ingestion.
2. **Acquisition** runs as a server pool, with each machine picking up queued messages which initiate an acquisition of data. Acquisition servers can act as a client for server protocols, process data sent via Portserver or from the UI servers.
3. **Controller** is a service that operates an acquisition schedule, sends out all user notifications and runs miscellaneous scheduled tasks.
4. **Extraction** is a server pool that responds to all data extraction requests.

5. **Processing** is a server pool that drives the Processing and Logic module.

User interface web servers

The user interface of eagle.io is driven by multiple application servers. These are load balanced using AWS, so that requests can always be fulfilled by multiple redundant servers, which are automatically scaled to meet current load requirements.

High availability datastore

The eagle.io database uses MongoDB Atlas for maximum resilience and data integrity. All data is stored here including user, configuration and time series data.

API gateway

Eagle.io exposes a RESTful API gateway to allow integration with third party web services. This component forms part of the UI servers.

Internal communication channel

A network communications channel ties all backend components together. Each server can read and create messages on the channel.

3 Reliability

A system is only as good as it is reliable; we have built multiple layers of redundancy to guard against data loss and ensure availability.

3.1 Application database

Eagle.io utilises a managed highly-available MongoDB cluster. The MongoDB database storage is designed to achieve a minimum of 99.9999999% durability with distributed fault tolerance and automated data recovery.

Each cluster is built for resilience with a multiple node replica set distributed across availability zones within a cloud region. This is further protected against regional and cloud outages with cross-region and cross-cloud failover.

3.2 File storage

Eagle.io utilises Amazon S3 to store files. Amazon S3 file storage data is designed to achieve a 99.999999999% durability and to sustain concurrent loss of data in two facilities.

Objects are redundantly stored on multiple devices across multiple facilities in an Amazon S3 Region.

4 Incident management

We have policies and procedures to ensure a consistent and effective approach to managing information security incidents (in the unlikely event that they occur), including communication of security events and weaknesses to appropriate stakeholders.

Our Security incident response team (SIRT) is responsible for:

- conducting periodic risk assessments to identify information security risks and determine the likelihood, impact, response and mitigation strategies
- conducting periodic reviews of the incident management procedure
- developing proactive controls to reduce the number of information security incidents
- identifying and responding to incidents

The incident response policies and processes are audited annually as part of our compliance processes and internally reviewed bi-annually.

5 Business continuity

As part of our business continuity management system, we have identified and quantified potential business risks that eagle.io may face in day to day operations. We have developed processes and procedures to mitigate the likelihood of risk events occurring, and documented contingency plans to be actioned to mitigate the impact of high consequence events (like catastrophic destruction of an AWS cluster).

The risk management plan considers events and impacts that pose both acute risks to business continuity (e.g., fire/earthquake/flood), and chronic risks (e.g., failure of corporate strategy).

The system includes the internal operational procedure for development and maintenance of the risk management plan, as well as the outputs of this procedure.

As part of our compliance processes, we must revisit our business continuity plans at a minimum once every 3 years. We go a step further and review our business continuity plan on an annual basis (or upon significant organizational and environmental change) and make changes to our internal policies & documentation as a result.

6 Disaster recovery

We maintain a disaster response and recovery plan documenting the actions, roles and responsibilities to restore services and address security risks that may arise from a disaster impacting our services. Events that are deemed to warrant contingency planning are those that:

- Have acute impacts that risk the ongoing viability of the business,

- Have complex or multi-step methods of recovery that would benefit from pre-planning.

6.1 Application

The application infrastructure utilises AWS CDK to perform all infrastructure configuration and code deployment changes. This allows the entire application to be rebuilt in a new availability zone if required.

6.2 Service providers

All organisations that provide services to eagle.io are reviewed at a minimum annually for sufficient security controls.

Our managed service provider for processing and storage, Amazon Web Services (AWS), is responsible for the logical and network security of our services provided through their infrastructure. More details about AWS compliance can be found here.

The MongoDB Atlas database service ensures data is backed up off site for disaster recovery purposes. Backup architecture consists of snapshots for long-term archival and compliance. More details about MongoDB Atlas compliance can be found here.

7 Malware scanning

Eagle.io system components are primarily provisioned using managed services, whose infrastructure is managed by Amazon Web Services. Threat & vulnerability management for those applications is managed via automated tools that review our source code, and by human review. Employees are required to use default operating system anti-virus applications running when using company provisioned hardware.

We have configured alerts to notify us to anomalous network or egress traffic behavior. We also run Amazon GuardDuty for intrusion detection.

Anomalous user activity, such as suspicious logins, is automatically flagged for internal review and notifications are sent out/sighted by the engineering team. The cloud services we use provide alerting for anomalous access & failed access attempts, these are human reviewed as required. Application error and uptime notifications are also reviewed as needed.

8 Product features related to security and control

Eagle.io provides a number of features for administrators and users to manage their account security. Below is a sampling of features available.

8.1 Admin management features

8.1.1 Privileged users

- The user that signed up for the account is assigned an "Owner" role. Only this user can close the account or change the subscription plan. The user has full unrestricted access to all assets and account settings including Managed accounts.
- A user can be added as an Administrator of an account which provides full unrestricted access to all assets and account settings excluding closing the account or changing the subscription plan. Administrators also have full access to Managed accounts.

8.1.2 Groups

Groups are used to organize your users and provide a convenient way to manage Workspace access and notifications.

8.1.3 Roles

Roles are assigned when Sharing a Workspace to restrict user and group access to certain features. Create unlimited new roles, or change existing role permissions as needed. Removing a role that has been assigned will restrict associated users to View only access.

8.1.4 Permissions

The following permissions are available in the system:

- View - Minimum required permissions to log in and view content.
- Read attachments - Preview and Download attachments.
- Export data - Export historic data associated with parameters and locations.
- Send messages - Send messages to users of the Workspace.
- Acknowledge alarms - Acknowledge alarms with or without comment.
- Edit states & alarms - Configure Node states and alarms, or Clear alarms.
- Edit quality & annotations - Assign quality and annotations to time series records.
- Edit data - Modify or remove time series record values.
- Operate - User can Acquire and Control Datasources and Parameters.
- Subscribe notifications - Subscribe to Nodes and receive Email and SMS notifications.
- Manage notifications - Manage user notification subscriptions on behalf of other users.
- Configure - Create, Modify & Delete Nodes and associated configuration or Upload Attachments.
- Security - Workspace Sharing and assignment of User Roles.

8.2 User provisioning and identity management

8.2.1 Authentication

Every interaction with the system is authenticated to ensure the user performing the interaction or operation has permission to do so.

8.2.2 API keys
API keys authenticate eagle.io HTTP API requests. The API keys section allows you to create, delete or modify the permissions for API keys associated with your user account. Multiple API keys can be active at once.

API resource requests are restricted to the assets within the associated account and are assigned an access level and permission to restrict the actions an API request can perform.

8.2.3 Two-Factor Authentication
Two-Factor authentication adds an extra layer of security to your user profile. This feature is optional, and can be enabled by the user. Once enabled by a user, they will need to provide a code generated by a supported 2FA tool along with their username and password when logging in.

8.3 User management

8.3.1 Workspaces
Individual users or groups can be added to a Workspace and assigned a security role specific to the Workspace (role-based security). The security roles are maintained by administrators and the owner. Each role has a set of security permissions assigned (e.g. acknowledge alarms, export data, configure) which define the interactions allowed by a user or group for all assets (nodes) in the Workspace.

You can restrict user and group access to any Node in the Workspace by opening the corresponding Node Security dialog and assigning the No Access role.

8.3.2 Sharing access
Users can be individually added to the workspace by entering the email addresses of the new users (separate multiple email addresses with a comma), choosing an optional expiry time, assigning a security role and clicking Add. You can remove or modify security roles for existing Workspace users. You can also modify the expiry time for existing Workspace users.

You can also add a pre-configured group of users to the Workspace, choose an optional expiry time, and assign a security role that will apply to all users in that group.

An email notification will be sent to users when they are added or removed from the Workspace or when their security role is changed. New users must follow the instructions in the email to configure their user profile and log in. Existing users who are currently logged in and viewing the interface will notice the new settings take effect immediately.

8.4 Visibility

The Events View provides an audit trail for all significant user interactions and system events. Table 1 details the events that are recorded in the feed.

Table 1 - System Events

Event Category	Event Node Type	Event Trigger Action
Acquisition	Source	• User operate action • Scheduled collection • Device connection • File received • API
Configuration	All	• User configuration change • API
State Change	Parameter	• Data acquisition
Notification	Source, Parameter, Report	• Alarm raised/cleared • Report generated on schedule
Control	Control Parameter	• User operate action • AP
Security	All	• User security change • API

Examples of user triggered events are:

- User login and logout
- User/Group added to workspace
- User/Group removed from workspace
- User/Group security role changed on workspace
- User/Group access expiry time changed on workspace
- User workspace sharing invite resent
- User/Group subscribed/unsubscribed to/from node
- Node created
- Node moved
- Node deleted
- Node configuration updated
- Node historic data modified
- Node historic data deleted
- Node historic data imported by user
- Node events cleared
- Node operation blocked due to account restrictions/capacity
- Node alarm acknowledged, cleared
- Account configuration updated
- Account security (administrators, groups, roles, apiKeys) updated

9 Application Security

Eagle.io can be accessed through three interfaces. Each has security settings and features that process and protect user data while ensuring ease of access.

9.1 User interfaces

9.1.1 Web

This interface can be accessed through any modern web browser. All Web requests must be made via a secure connection (HTTPS).

9.1.2 Mobile

Eagle.io can be accessed on a mobile phone through any modern web browser. It has been built on HTML5 technology, which ensures the platform wraps items on the website to display on desktop or handheld devices. All Mobile requests must be made via a secure connection (HTTPS).

9.1.3 API

The HTTP API has been designed to make it easy for developers to interact with eagle.io from a 3rd party application or service. Authentication to the eagle.io API is achieved by providing one of your API keys in the X-Api-Key request header. All API requests must be made via a secure connection (HTTPS).

9.2 Data integrity

Our software is built to check the validity of data input prior to ingestion and to sanitize API outputs. This is checked and audited via a combination of end-to-end tests, and manual testing. All storage is backed up to ensure minimal data loss in the event of an integrity error.

9.3 Data retention

Data in our MongoDB is stored indefinitely for the duration of the subscription. The data is deleted from the system 30 days after account termination.

Data is aged out of our rolling back-up after 6 months.

9.4 Payment Card Industry (PCI) Compliance

Our payment processor is a validated PCI DSS (Level 1) Compliant Service Provider and is on Visa's Global Compliant Provider List and MasterCard's SDP List.

We never store our customers' credit card numbers, these are handled by our payment processor.

10 Encryption

10.1 Data in transit

To protect data in transit Eagle.io uses Secure Sockets Layer (SSL)/Transport Layer Security (TLS) for data transfer. All communication between our users and us is conducted in a highly secure fashion using the TLS 1.2 Protocol, a 2048 bit RSA key, and the ECDH 256 bit cipher suite.

10.2 Data at rest

Our MongoDB managed services provider enables encryption for data at rest using encrypted storage volumes.

10.3 Key management

10.3.1 Internal SSH keys

Access to critical production systems is restricted with unique SSH key pairs. Security policies and procedures require protection of SSH keys:

- SSH keys are securely generated on a local machine and stored with the correct permissions set - the storage folder must not be writeable to anyone else and none of the secret keys are readable to anyone else.
- Private SSH keys are not shared.
- SSH keys no longer in use are destroyed and removed from servers. This includes when staff or contractors leave eagle.io and access is revoked.

10.4 Protecting authentication data

User account passwords are never stored as plain-text in the database, but are salted and hashed using a slow hash function to increase security.

11 Network Security

Our network has been set up with minimal access to outside networks. All access to our protected network is achieved via our VPN, which operates using a secure encrypted tunnel from whitelisted IPs.

This architecture design protects against "man in the middle" attacks (intercepting communications between our employees' devices and the wireless router), as there would be no information from our protected network exposed.

Similarly, the office wireless network would not allow access to our protected network as that is accessed only via VPN.

We utilize AWS security groups to protect network/system environments and use segmented networks so only servers that work together can communicate with each other.

A web application firewall is integrated to protect the system against common web exploits and bots that may affect availability, compromise security, or consume excessive resources.

12 Vulnerability management

We maintain secure patching and software updates of all our systems, including static application security testing (SAST) and dynamic application security testing (DAST). We automate the detection and updating of vulnerable dependencies as part of our Continuous Integration (CI) pipeline. Anything that cannot be automatically merged is prioritized for remediation by our engineering team. All security defects found during review or testing are remediated prior to deployment to production. Internally, we treat these as a top priority.

We automate the deployment of software and infrastructure used to deliver our services. Customers are not able to install software on eagle.io; however, they are able to provide some limited scripts for processing and logic. The processing and logic scripts are run in individual script sandboxes that are heavily restricted. We also run amazon GuardDuty which provides intelligent threat detection at a platform level.

13 Change management

Eagle.io has a written procedure that sets out the method of managing the process to make changes to the platform; both application changes and changes to the infrastructure. Conformance with this procedure is audited annually.

13.1 Codebase

All of our source code is stored on cloud based source control providers. Only approved staff members are granted access to private repositories. We have processes enabled via our cloud based source control provider to automatically detect and revoke access keys in the case of any leak.

13.2 Secure system engineering and coding principles

Eagle.io requires that its developers write code that is in line with the secure coding practices outlined in v2 of the OWASP Foundation's Secure Coding Practices guide. This ensures that common web application security risks, including the OWASP top 10 Web Application Security Risks are mitigated. The OWASP Coding practices guide covers areas of secure application development including:

- Input Validation
- Output Encoding
- Authentication and Password Management
- Session Management

- Access Control
- Cryptographic Practices
- Error Handling and Logging
- Data Protection
- Communication Security
- System Configuration
- Database Security
- File Management
- Memory Management
- General Coding Practices

If at any point during the development lifecycle, a developer or reviewer notes a violation of the OWASP secure coding practice guidelines, then a fix is prioritized as a change request or fixed immediately if the code in question is related to the current change that the developer is working on or reviewing. Upon noting a violation of the OWASP guidelines, the developer will assess whether there was potential for a security incident to have occurred related to the guideline violation. If the assessment shows a high likelihood of a security incident, then a request to investigate the incident is raised to the team lead or product owner, and the investigation follows eagle.io's standard security incident investigation and management procedures.

13.3 Development model

Eagle.io adopts an agile and iterative development model. Multiple development and testing iterations are to be conducted in the sprint. Developers are required to adhere to industry best practices for writing and structuring code and writing tests. Eagle.io developers typically write a combination of unit and integration tests for new feature development and bug fixes.

When reviewing code, the reviewer assesses compliance with product requirements prior to issuing authorisation in consideration of the following:

- Conformance with industry best practices for software development
- Correct structure and placement of code in the overall system
- Presence of bugs in programming logic
- Any Information security risks that may be introduced
- Conformance with eagle.io's secure coding practices

Should non-conformance be identified, the task or module is resubmitted to the development team with notes detailing the non-conformance.

14 Scanning and Security penetration testing

We perform automated and manual application security testing on a regular basis to identify and patch potential security vulnerabilities and bugs.

Additionally, we contract with a reputable third-party vendor to perform annual penetration testing.

15 Information security framework

Eagle.io is merging and consolidating its ISO 27001 program into Bentley Systems Corporate ISO 27001 (information security) and ISO 27701 (privacy) programs. Completion of this is expected by Q4 2024.

15.1 Policies and procedures

The framework policy objectives are:

- Provide a framework for the design, implementation and management of an effective ISMS which ensures that the system's information assets are properly identified, recorded, and afforded suitable protection at all times.
- Ensure that all users and staff are aware of their responsibility for the security of information and comply with all security policies.
- Ensure confidentiality, integrity, and availability of the business's information assets.
- Ensure that all vulnerabilities, threats and risks to information assets are formally identified, understood, assessed and controlled in accordance with the Risk Assessment Methodology.

The following documents make up the foundation of the Information Security Management System (ISMS).

15.1.1 Access Control Policy

The Access Control Policy aims to prevent unauthorised access to information and information processing facilities. Processes are defined to guide how access controls are applied by eagle.io, covering all stages in the lifecycle of user access, from the initial registration process of new users to the final deregistration of users who no longer require access to information systems and services.

15.1.2 Incident management procedure

This procedure ensures a consistent and effective approach to the management of information security incidents, including communication on security events and weaknesses, detection, escalation and reporting.

15.1.3 Business Continuity Plan

This operational procedure describes the processes undertaken by eagle.io to manage the risk to the system in the event of a crisis/disaster.

15.1.4 Change Management

This procedure sets out the method of managing the process to make changes to the eagle.io platform; both application changes and changes to the architecture.

15.1.5 Human resource security policy

The objective of this policy is to define rules and guidelines that apply before, during and after employment of all eagle.io staff. All of our employees undergo training on relevant security matters that relate to their job.

15.1.6 Information classification and handling policy

This policy aims to provide a process to assess the data that eagle.io holds and the level of protection it should be given in accordance with its importance.

15.1.7 Operational Procedure: Project Management

This operational procedure describes the process undertaken by development staff employed by eagle.io when undertaking project work.

15.1.8 Operating Procedures for IT Management

This procedure sets out the method of managing the ongoing management of the infrastructure and IT elements of eagle.io.

15.1.9 Acceptable Use Policy

The purpose of this policy is to outline the acceptable use of information assets at eagle.io. Inappropriate use of such systems may expose eagle.io to risks, including virus attacks, compromise of network systems and services, and legal issues.

16 Physical security

16.1 Physical security

We are a remote first company but do have offices. Each office has a locked door preventing entry that only authorised people have the key for.

16.2 Visitor and access policy

Physical access to facilities, other than public entrances and areas, is restricted to authorized personnel and visitors who are accompanied by eagle.io representatives.

16.3 Infrastructure

While we have offices, they do not house a data center or any other information systems. Platform data is stored on AWS. AWS maintains stringent security standards. For more information please refer to the AWS customer data center controls documentation.

17 Compliance

Eagle.io has held ISO 27001:2013 (Information Security Management Systems) certification until March 2024. Eagle.io is currently undergoing a migration program to be incorporated under Bentley Systems ISO 27001 certification. Eagle.io's inclusion in Bentley's ISO system will allow the efficient use of the larger security infrastructure in Bentley to identify and manage risks.

We are continually seeking to enhance our already robust security and compliance framework. We are currently undergoing assessment for inclusion in the Security, Trust and Assurance Registry of the Cloud Security Alliance, which certifies cloud provider trust and assurance.

18 Privacy

Our privacy policy is available at https://eagle.io/policies/privacy and provides details on the following:

- What information do we collect?
- What do we use your information for?
- How do we protect your information?
- Do we use cookies?
- Do we disclose any information to outside parties?
- Terms of Use
- What information do we collect?
- Changes to our Privacy Policy

19 Summary

Eagle.io offers simple tools to help turn time-series data into actionable intelligence while providing the security measures and compliance certifications organizations require. With a multi-layered approach that combines a robust back-end infrastructure with a comprehensive Information Security Framework, we provide system integrators and consultants a powerful solution that can be tailored to their unique needs. To learn more about eagle.io, contact our sales team at sales@eagle.io.