Data Privacy & PII Redaction

Safeguarding Data Privacy

Lightrun offers multiple built-in mechanisms for ensuring that sensitive data is not exposed to unauthorized individuals, keeping your customers privacy and safety

PII redaction
Prevent sensitive data from appearing in Lightrun actions - snapshots and dynamic logs

Blocklists
Prevent developers from inserting Lightrun actions inside sensitive code areas

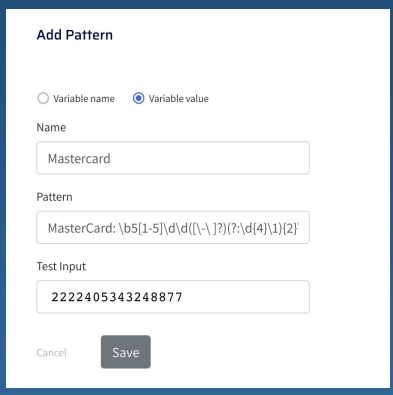
PII Redaction

Lightrun provides a number of tools which account administrators use to ensure that sensitive data is not exposed via use of Lightrun. The approach is based on the shared responsibility model, familiar from public cloud: Lightrun ensures that strong controls are available and respected, while it is the customer's responsibility to identify sensitive data and ensure that filters are created as needed.

Below we describe the filters which are applied to all data subject to capture via Lightrun. These filters will redact sensitive data to ensure that neither Lightrun staff nor the customer's developers are exposed to sensitive data, either purposefully or accidentally. Lightrun Solution Engineering collaborates closely with our customers to create PII filters which match and redact the sensitive data identified by the customer. PII redaction can be used to prevent developers from capturing sensitive data in snapshots and dynamic logs.

There are two options for redacting sensitive data

Variable name
Data is redacted based on variable name. Any variables which match the supplied patterns will be excluded from the data Lightrun actions capture. For example, adding the pattern `apiToken` will prevent Lightrun from logging data from any variable which includes `apiToken` in the variable name.
So, for example, variables `my_apiToken`, `theOtherapiToken`, and `someapiTokenVariable` will all be redacted.



- **Variable value**
  Data is redacted based on matching a specified regular expression pattern. The regex pattern is matched to a variable value. For example, the following regex pattern `\b5[1-5]\d\d([\-\ ]?)(?:\d{4}\1){2}\d{4}\b` will redact all Mastercard debit/credit card data from Lightrun.

PII Redaction Best Practices

To ensure user privacy and improve data security, it is important that Personally Identifiable Information (PII) is not included in snapshots or logged in IDE plugins. This includes information such as email addresses, credit card numbers, passwords and other secrets, personal mobile numbers, social security numbers, etc. When implementing PII redaction with Lightrun, we recommend using the following best practices to reduce the risk of exposing sensitive data through Lightrun.

- **Access Limitation**

  Ensure that only trusted users in your organization have access to specific roles and permissions in Lightrun. Only users with `role_manager` or System `administrator permissions` can specify or remove PII redaction patterns from Lightrun.

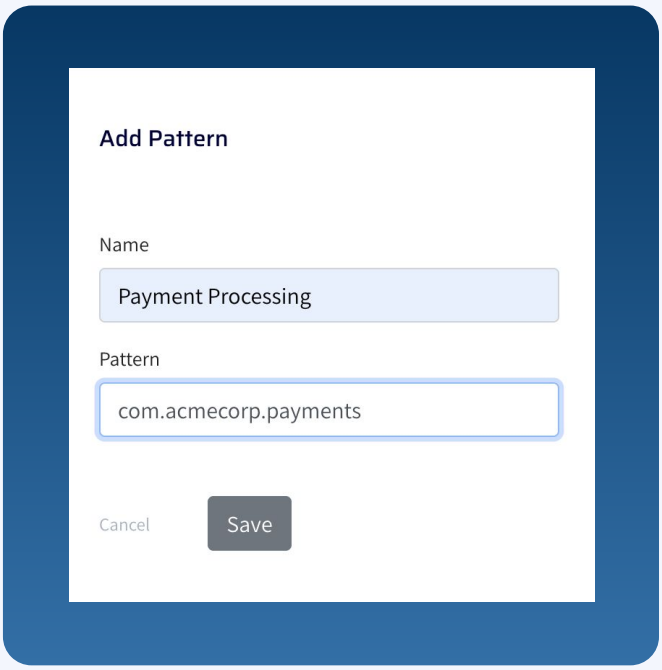
- **Combine PII redaction rules for both variable values and variable names to protect sensitive information**.

  Specifying both the variable names and values helps to reduce the risk of sensitive data being exposed to Lightrun. The combination of both methods can ensure that developers cannot add Lightrun actions containing expressions which expose sensitive data.

Blocklisting

Blocklists can be used to prevent snapshots from being inserted in classes that might expose sensitive data. Files and packages that include the patterns you've specified in the Blocklist table are protected and your team won't be able to add snapshots into those code areas.

You can configure blocklists to include package and class names, file names, and directory paths. You can also add blocklist exceptions for any relevant subclasses in which you want to allow snapshot insertion. Each time your application is started, the agent's blocklist configuration is downloaded and applied to all future actions. Lightrun agents will fetch updated blocklists when they start up. To apply new filters to your existing agents you'll need to restart those agents.



Audit Trail Capabilities

Lightrun maintains a record of your organization's Lightrun system usage, which is crucial for observing continuous compliance, performing system audits, and maintaining security. The stored audit logs include data about activities related to the Management Portal, Lightrun plugins, and agents.

With the Lightrun audit logs, you can answer questions such as:

Which user created an action (dynamic log, snapshot etc.) and when?

On which source file name and line actions were placed?

Which changes have been made to your organization's account, and when?

Audit log events are covering the following domains and actions

- Lightrun actions - creation (success, failure)
- Users - creation, deletion, authentication
- Agents - connecting, disconnecting
- Tags - Creation
- Integrations - creation

Each event contains a rich set of metadata, for example, an action metadata contains the following details

- Action type - log, snapshot, metric
- Action id - unique action identifier
- File name - source file name
- Line - source file line
- Condition
- Expression
- Max hits count
- Ignore quota

Audit events can be consumed either by

- Viewing them from the Lightrun management portal (requires permissions)
- Exporting the data to an S3 bucket in Syslog format
- Retrieving and processing them via API

The audit logs exports in S3 buckets are updated daily and have a default retention period of 12 months.

Future Data Privacy Roadmap

In response to feedback from our customers, Lightrun continually works to strengthen our data protection capabilities. In this context, we are investigating the option to add a new feature which will enable adding managerial supervision over action creation.

Similar to the way a pull request mechanism is used for approving code changes, we envision a mechanism for observability actions approval. Such a mechanism means that new actions created by developers will require a review and approval by another user (for example a team leader) and only then they will be added to the target environment. The user reviewing the action will be able to validate that it does not expose any sensitive data by reviewing the action metadata (source file, line, expression). Approval will be done using the Lightrun product or by integrating to a 3rd party system.

Strategies for Secure Dynamic Log Points

We recommend using the following measures for keeping data privacy and safeguarding your customers' sensitive information:

- **Manage developers' access to runtime environments, such as production, using Lightrun role-based access control**
  - Ensure that only authorized individuals can access environments which contain sensitive user data
  - Create separate agent pools for environments which contain sensitive users data and use them when applying access controls
- **Connect Lightrun with your identity management system and leverage the SAML and SCIM integrations to ensure that only authorized developers have access to Lightrun**
- **Configure PII redaction rules for both variable names and values to make sure that sensitive data is not accessible to Lightrun actions**

  Use standard regular expression patterns for redacting sensitive information such credit card numbers, secrets such as tokens and API keys, email addresses and social security numbers
- **Add blocklist rules for preventing developers from adding Lightrun actions in sensitive code sections, for example one which deals with payment processing or user's personal details**
- **Routinely review Lightrun audit logs to identify any instances where an attempt was made to log sensitive data.**

  Consider forwarding the audit logs to you SIEM solution

Lightrun Data Privacy

Lightrun Deployment Options

Lightrun offers multiple deployment options:

SaaS

Lightrun hosts the server components in the highly-secure and available AWS environment, based on hardened virtual servers and services. Lightrun's Engineering and DevOps teams are responsible for the ongoing maintenance and uptime of the environment.

- Multi tenant - infrastructure is shared with other customers
- Single tenant - dedicated infrastructure

Self hosted

Lightrun is installed within your organizational network or through a private cloud, via Docker or Kubernetes. The customer's IT or DevOps team is responsible for the ongoing maintenance, as is the case for any other internal/local resource. In addition, all of the customer's existing organizational security controls and policies automatically apply to all of Lightrun's components.

Do I need to declare Lightrun as a sub-processor?

It depends how you deploy and use Lightrun.

There is no need to declare Lightrun as a subprocessor in the following scenario:

- Lightrun is deployed in a self-managed / on-premise deployment model
- Lightrun is not deployed on services which process customer data
- You restrict the delivery of Lightrun action data to the local filesystem and your existing log-processing tooling (such as APMs), with no action data sent to the Lightrun Server or to developer plugins.

Data Privacy Comparison

Data

Self Hosted

No customer data leaves the customer premises or is sent to Lightrun

SaaS

Action data is sent to Lightrun as a gateway to transmission to developers, and so that customer-appointed account managers have the opportunity to oversee use of the system.

The Lightrun server stores this data for a brief period of time which is configurable and can be reduced by customers to just the bare minimum required to transfer the data to the developers who requested it.
There is an option to limit this behavior by sending Lightrun logs to Stdout only, meaning they are not being sent to the IDE plugin or management portal

An admin can control the expiration period for Lightrun action data stored on the Lightrun server

Source code

Lightrun utilizes the developers' existing IDE and therefore the customer's source code does not leave its infrastructure. Only the customer has access to the source code, at all times

Encryption

The communication between all Lightrun components and the Management server is always established over industry-standard TLS 1.2 encrypted channels.
Certificate pinning is utilized both in the agent and the client

All customers' data is stored in a self managed Database. We strongly recommend to encrypt the database using industry standards

All customers' data hosted in AWS is encrypted using Amazon's AES-256 encryption algorithm and stored on Elastic Block Store (EBS) storage and Relational Database Service (RDS) databases

For more information regarding Lightrun security please see our security whitepaper