Cortex XSIAM Premium Documentation

1 | Data Collection

Abstract

Data can be collected both from Palo Alto Networks products, and from third-party vendor products.

Learn about the types of data that can be collected into your system, vendor support, and how to configure collection.

1.1 | Visibility of logs and issues from external sources

Abstract

Cortex XSIAM provides visibility into your external logs. The availability of logs and issues varies by the data source.

> **LICENSE TYPE:**
>
> Data collection may require an add-on.

The following table describes the visibility of each vendor and device type, and where you can view information ingested from external sources, depending on the data source.

A ✓ indicates support and a dash (—) indicates the feature is not supported.

Network connections

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Amazon S3 (flow logs)	✓ Raw data is searchable in XQL Search.	✓ Option to ingest network flow logs as Cortex XSIAM network connection stories that are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Amazon S3 (Route 53 logs)	✓ Raw data is searchable in XQL Search.	✓ Option to ingest network Route 53 DNS logs as Cortex XSIAM network connection stories that are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Azure Event Hub	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from flow logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Azure Network Watcher (flow logs)	✓ Raw data is searchable in XQL Search.	✓ Option to ingest network flow logs as Cortex XSIAM network connection stories that are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from flow logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Check Point FW1/VPN1	✓ Raw data is searchable in XQL Search. **NOTE:** Logs with `sessionid = 0` are dropped.	✓ Network stories that include Check Point network connection logs are searchable in the Query Builder and in XQL Search. **NOTE:** Logs with `sessionid = 0` are dropped.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	✓ Alerts from Check Point firewalls are generated throughout Cortex XSIAM when relevant.
Corelight Zeek	✓ Raw data is searchable in XQL Search.	✓ Network stories that include Corelight Zeek network connection logs are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Cisco ASA and Cisco AnyConnect VPN	✓ Raw data is searchable in XQL Search.	✓ Network stories that include Cisco network connection logs are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Fortinet Fortigate	✓ Raw data is searchable in XQL Search.	✓ Network stories that include Fortinet network connection logs are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	✓ Alerts from Fortinet firewalls are generated throughout Cortex XSIAM when relevant.
Google Cloud Platform (flow logs, DNS logs)	✓ Raw data is searchable in XQL Search.	✓ Option to ingest network flow logs as Cortex XSIAM network connection stories and Google Cloud DNS logs that are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Okta	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** • While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs. • IOCs and BIOCs are only generated for these event types: **`sso`** and **`session_start`**.	—
Prisma Access Browser	✓ Raw data is searchable in XQL Search.	—	—	—
Windows DHCP via Elasticsearch Filebeat	✓ Raw data is searchable in XQL Search.	✓ Cortex XSIAM uses Windows DHCP logs to enrich your network logs with hostnames and MAC addresses that are searchable in XQL Search.	—	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Zscaler Internet Access (ZIA)	✓ Raw data is searchable in XQL Search.	✓ Network stories that include ZIA network connection and firewall logs are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** • While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs. • Analytics, IOCs and BIOCs are only generated on the Firewall data. • The Zscaler Nanolog Streaming Service (NSS) feed for web logs is only used for Correlation Rules and threat hunting.	—
Zscaler Private Access (ZPA)	✓ Raw data is searchable in XQL Search.	✓ Network stories that include ZPA network connection logs are searchable in the Query Builder and in XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** • While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs. • The Zscaler Nanolog Streaming Service (NSS) feed for web logs is only used for Correlation Rules and threat hunting.	—

Authentication services/Audit logs

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Amazon S3 (audit logs)	✓ Logs and stories are searchable in XQL Search	✓ Option to stitch audit logs with authentication stories that are searchable in the Query Builder and XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Azure Event Hub (audit logs, AKS logs)	✓ Logs and stories are searchable in XQL Search	✓ Option to stitch audit logs with authentication stories that are searchable in the Query Builder and XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Google Cloud Platform (audit logs, GKE logs)	✓ Raw data is searchable in XQL Search.	✓ Option to stitch audit logs with authentication stories that are searchable in the Query Builder and XQL Search.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Google Workspace	✓ Raw data is searchable in XQL Search.	✓ Relevant Login, Token, Google drive, SAML, Admin Console, Enterprise Groups, and Rules audit logs normalized into authentication stories. All are searchable in the Query Builder.	✓ For all logs, Cortex XSIAM can generate Cortex XSIAM issues (Analytics and Correlation Rules) when relevant from logs.	—
Microsoft 365 email	✓ Email logs are searchable in XQL Search	✓ Microsoft 365 normalized email stories	✓ For Microsoft 365 email logs, Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from the email logs.	—
Microsoft Office 365	✓ Logs and stories (Azure AD authentication and audit logs only) are searchable in XQL Search	✓ Azure AD authentication logs and Azure AD Sign-in logs normalized into authentication stories. Azure AD audit logs normalized to cloud audit logs stories. Exchange Online, SharePoint Online, and General audit logs normalized into stories. All are searchable in the Query Builder.	✓ For all Microsoft Office 365 logs, Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from Office 365 logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Okta	✓ Logs and stories are searchable in XQL Search	✓ Logs stitched with authentication stories are searchable in the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules only) when relevant from logs. **NOTE:** • While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs. • IOCs and BIOCs are only generated for these event types: **`sso`** and **`session_start`**.	—
OneLogin	✓ Raw data is searchable in XQL Search.	✓ All log types are normalized into authentication stories, and are searchable in the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
PingFederate	✓ Logs and stories are searchable in XQL Search	✓ Logs stitched with authentication stories are searchable in the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
PingOne for Enterprise	✓ Raw data is searchable in XQL Search	✓ Logs stitched with authentication stories are searchable in the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Operation and system logs from cloud providers

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Amazon S3 (generic logs)	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Amazon CloudWatch (generic logs, EKS logs)	✓ Raw data is searchable in XQL Search.	✓	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Azure Event Hub	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Google Cloud Platform	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Google Kubernetes Engine	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Okta	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
CrowdStrike Falcon Data Replicator and CrowdStrike Streaming API (collection from APIs)	✓ Raw data in `crowdstrike_falcon_incident_raw` and `crowdstrike_fdr_raw` is searchable in XQL search.	✓ CrowdStrike logs are normalized into `xdr_data` and are searchable using XQL search and the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Microsoft Defender for Endpoint	✓ Raw data in `msft_defender_raw` is searchable in XQL search.	✓ Microsoft Defender logs are normalized into `xdr_data` and are searchable using XQL search and the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
SentinelOne DeepVisibility	✓ Raw data in `sentinelone_deep_visibility_raw` is searchable in XQL search.	✓ SentinelOne DeepVisibility logs are normalized into `xdr_data` and are searchable using XQL search and the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Windows Event Collector	✓ Windows event logs are available with agent EDR data and are searchable in XQL Search. The normalized Windows event log data is also available in `microsoft_windows_raw` and is searchable in XQL Search.	✓ Windows event logs are stitched with agent EDR data and are searchable in the Query Builder. The Windows event logs are also normalized into the common Cortex Windows event format in `microsoft_windows_raw` and are searchable using the Query Builder.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Custom external sources

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Any Vendor Sending CEF, LEEF, CISCO, CORELIGHT, or RAW formatted Syslog	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	✓ To enable Cortex XSIAM to display alerts from other vendors, you must map your alert fields to the Cortex XSIAM field format (see Ingest external alerts).
Any vendor CSV files on a shared Windows directory	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Any vendor logs stored in a database	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Any vendor logs stored in files on a network share	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Any vendor logs from a third party source over FTP, FTPS, or SFTP	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Any vendor sending NetFlow flow records	✓ Raw data is searchable in XQL Search.	✓ NetFlow events are stitched with the Agent's EDR data and other Network products to a Session Story, and are searchable in the Query Builder and in XQL.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Any vendor sending logs over HTTP	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	✓ To enable Cortex XSIAM to display alerts from other vendors, you must map your alert fields to the Cortex XSIAM field format (see Ingest external alerts).
Apache Kafka	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
BeyondTrust Privilege Management Cloud	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Box	✓ Raw data is searchable in XQL Search.	✓ Selected Box audit event logs are normalized into stories and are searchable in the Query Builder and in XQL.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Dropbox	✓ Raw data is searchable in XQL Search.	✓ Selected Box audit event logs are normalized into stories and are searchable in the Query Builder and in XQL.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Elasticsearch Filebeat	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Forcepoint DLP	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
IoT Security	✓ Raw data is searchable in XQL Search.	✓ Cortex XSIAM uses IoT Security information to improve analytics detection and assets management information.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	—
Proofpoint Targeted Attack Protection	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAMCortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Salesforce.com	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAM can generate Cortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
ServiceNow CMDB	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAMCortex XSIAM issues (Correlation Rules only) when relevant from logs.	—

Vendor And Device Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility	Vendor Alert Visibility
Strata Logging Service	✓ Raw data is searchable in XQL Search.	✓ Detection events are stitched with other Palo Alto Networks product logs to stories, and are searchable in the Query Builder and in XQL.	✓ Cortex XSIAM can generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC and BIOC issues are only generated on normalized logs.	✓ Alerts from NGFW are generated throughout Cortex XSIAM when relevant.
Workday	✓ Raw data is searchable in XQL Search.	—	✓ Cortex XSIAMCortex XSIAM issues (Correlation Rules only) when relevant from logs.	—
Any vendor sending alerts	—	—	—	✓ Alerts are surfaced throughout Cortex XSIAM when relevant. To enable Cortex XSIAM to display your alerts, you must map your alert fields to the Cortex XSIAM field format (see Ingest external alerts).

Datasets created from ingesting data

When ingesting data from an external source, Cortex XSIAM creates a dataset that you can query using Cortex Query Language (XQL). Datasets created in this way use the following naming convention.

```
<vendor_name>_<product_name>_raw
```

For example: cisco_asa_raw

The datatypes used for the fields in an imported dataset are automatically assigned based on the input content. Fields can have a datatype of `string`, `int`, `float`, `time`, or `boolean`. All other fields are ingested as a JSON object.

For CEF type files, when extension values are quoted, the CEF parser automatically removes the quotes from the values. In addition, files containing invalid UTF-8 are parsed under XQL mapping field `_invalid_utf8`.

1.2 | Visibility of Cortex XSIAM audit and authentication logs

Abstract

Monitor Cortex XSIAM authentication and audit logs for detecting attacks on Cortex XSIAM.

You can audit and query Cortex XSIAM authentication logs and activity logs to track and create issues for malicious activity on Cortex XSIAM.

A ✓ indicates support and a dash (—) indicates the feature is not supported.

Log Type	Raw Data Visibility	Normalized Log Visibility	Cortex XSIAM Issue Visibility
Cortex XSIAM authentication logs	✓ Logs and stories are searchable in XQL Search.	✓ Cortex XSIAM authentication logs normalized into authentication stories, which are searchable in the Query Builder.	✓ Cortex XSIAM can create Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** Cortex XSIAM can create Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs.
Cortex XSIAM audit logs	✓ Logs and stories are searchable in XQL Search.	✓ Cortex XSIAM authentication logs are normalized into SaaS stories which are searchable in the Query Builder.	✓ Cortex XSIAM can create Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs. **NOTE:** Cortex XSIAM can create Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from logs.

1.3 | External data ingestion vendor support

Abstract

To augment your Cortex XSIAM data, you can set up Cortex XSIAM to ingest data from a variety of external third-party sources.

LICENSE TYPE:

Data collection may require an add-on.

To provide you with a more complete and detailed picture of the activity involved in a case, you can ingest data from a variety of external, third-party sources into Cortex XSIAM.

Cortex XSIAM can receive logs, or both logs and alerts, from the source. Depending on the data source, Cortex XSIAM can provide visibility into your external data in the form of:

- Log stitching with other logs in order to create network or authentication stories.
- Raw data in queries from XQL Search.
- Alerts reported by the vendor throughout Cortex XSIAM, such as in the issues table, cases, and views.
- Issues raised by Cortex XSIAM on log data, such as analytics findings.

To ingest data, you must set up the Syslog Collector applet on a Broker VM within your network.

The following table summarizes the vendor data that can be ingested, according to log or data type.

Log Or Data Type	Vendor Support
Network Connections	• Amazon S3 (flow logs) • Amazon S3 (Route 53 logs) • Azure Event Hub • Azure Network Watcher (flow logs) • Check Point FW1/VPN1 • Cisco ASA and Cisco AnyConnect VPN • Corelight Zeek • Fortinet Fortigate • Google Cloud Platform (flow logs, DNS logs) • Okta • Prisma Access Browser • Windows DHCP via Elasticsearch Filebeat • Zscaler Internet Access (ZIA) • Zscaler Private Access (ZPA)
Authentication Services/Audit Logs	• Amazon S3 (audit logs) • Azure Event Hub (audit logs, AKS logs) • Google Cloud Platform (audit logs, GKE logs) • Google Workspace • Microsoft 365 (email) • Microsoft Office 365 • Okta • OneLogin • PingFederate • PingOne for Enterprise
Operation and System Logs from Cloud Providers	• Amazon S3 (generic logs) • Amazon CloudWatch (generic logs, EKS logs) • Azure Event Hub • Google Cloud Platform • Google Kubernetes Engine • Okta
Endpoint Logs	• CrowdStrike Platform • Microsoft Defender for Endpoint • SentinelOne DeepVisibility • Windows Event Collector

Log Or Data Type	Vendor Support
Custom External Sources	• Any Vendor Sending CEF, LEEF, CISCO, CORELIGHT, or RAW formatted Syslog • Any vendor CSV files on a shared Windows directory • Any vendor logs stored in a database • Any vendor logs stored in files on a network share • Any vendor logs from a third party source over FTP, FTPS, or SFTP • Any vendor sending NetFlow flow records • Any vendor sending logs over HTTP • Apache Kafka • BeyondTrust Privilege Management Cloud • Box • Dropbox • Elasticsearch Filebeat • Forcepoint DLP • IoT Security • Proofpoint Targeted Attack Protection • Salesforce.com • ServiceNow CMDB • Strata Logging Service • Workday • Any vendor sending alerts

1.4 | Manage instances

You can manage the Instances configured for a Data Source on the Data Sources page. You can edit, delete, enable, or disable instances, and refresh log data.

1. Select Settings → Data Sources.
2. Find an instance by clicking on a Data Source name or using the Search field.
3. In the row for the instance name, take the required action:

Action	Instructions
Enable or disable an Instance	Select or deselect the Enable checkbox.
Refresh log data	Click the Refresh icon.

Action	Instructions
Edit an Instance	a. Click the Edit icon. b. In Edit Data Source, you can update the values in the Connect and Collect sections. The options under Recommended Content are view only. c. Test the configuration. d. Connect the updated Instance.
Delete an Instance	Click the Delete icon. If you delete all the instances for a Data Source, the Data Source is not listed on the Data Sources page.

1.4.1 | Add a new data source or instance

Abstract

Use the Data Source Onboarder to add a new data source or instance in Cortex XSIAM.

You can add a new data source with the Data Source Onboarder. The Onboarder installs the data source, sets up an instance, configures playbooks and scripts, and other recommended content. The Onboarder offers default (customizable) options and displays all configured content in a summary screen at the end of the process.

1. Select Settings → Data Sources.
2. Select one of the following options:
   - Add Data Source
   - Add New Instance for an integrated data source by clicking the menu in the right corner of an existing data source. Then skip to Step 4.
3. Select a data source to onboard and click Connect.

   Hovering over a data source displays information about the data source and its integrations. Data sources that are already integrated are highlighted green and show Connect Another Instance. To see details of existing integrations, click on the number of integrations.

   The data sources are drawn from the Marketplace, Custom Collectors, and integrations. If you search for a data source and No Data Sources Found, click Try searching the Marketplace, to view the marketplace page prefiltered for your search. If there are no available options in the Marketplace, you can use one of the Custom Collectors to build your own.

   > **NOTE:**
   >
   > - If a data source contains multiple integrations, the integration configured as the default integration will used by the Data Onboarder. The default integration of the content pack is indicated in each content pack's documentation. The other integrations are available for configuration in the Automation and Feed Integrations page after installing the content pack.
   > - Not all content packs are supported.
   > - When adding XDR data sources the Data Source Onboarder is not available, however, you can still enable the data source. Cortex XSIAM then creates an instance and lists it on the Data Sources page.
4. In the New Data Source window, complete the mandatory fields in the Connect section.

   For more information about the fields, click the question mark icon.
5. (Optional) Under Collect, select Fetched alerts and complete the fields.
6. Under Recommended Content, review and customize the options.

   The items in this section are content-specific. Some options are view only, and others are customizable. Click on each option for more information:
   - Classifiers & Mappers
   - Data Normalization: Parsing rules and data models
   - Correlations: Correlation rules included in the pack
   - Automation: Playbooks and Scripts included in the pack.

     You can select the Playbooks and Scripts that you want to enable. By default, recommended options are selected. Any unselected content is added as disabled content. Depending on the selected playbook, some scripts are mandatory.
   - Dashboards & Reports: Recommended dashboards, widgets, and reports

> **NOTE:**
>
> - If you are adding a new instance to an existing data source, these options are View only.
>
> You can adjust the view-only options on the relevant page in the system, for example Correlations, Playbooks, or Scripts.
>
> - Cortex XSIAM automatically installs content packs with required dependencies and updates any pre-installed optional content packs. You can also Select additional content packs with optional dependencies to be configured during connection.

7. Test the configuration.

   If the test fails, you can Run Test & Download Debug Log to debug the error.

8. Connect the data source.

9. Review the configuration in the summary screen.

   If errors occurred during the test, you can click See Details and Back to Edit to revise your configuration. For advanced configuration, click on any item to open a new window to the relevant page in the system (for example, Correlations or Playbooks) filtered by the configuration.

10. Click Finish to return to the Data Sources page.

1.4.2 | How to configure the scanning settings for supported services

Abstract

How to edit a cloud instance, including data asset types.

1. In Cortex XSIAM Command Center, in the lower left area, click Settings → Data Sources.
2. On the Data Sources screen, click a cloud service provider or other data source and then click the View Details link.
3. On the Cloud Instances screen, click an instance name link. A screen displaying the instance name opens.
4. At the bottom of the screen, under Accounts (AWS), Subscriptions (Azure), or Projects (GCP), right-click an item in the list and then in the context menu, select Edit.
5. In the screen that opens, under Data assets classification options, you can deselect any data asset types.

   > **NOTE:**
   >
   > All the asset types are selected by default. When you deselect a data asset type, it is not included when the system runs the next scan operation, reducing the total scan time.

6. Click Save.

   For more information about supported assets in Cortex Cloud Data Security, see Supported assets in Cortex Cloud Data Security.

1.4.3 | Manage cloud instances

Abstract

You can manage the cloud instances configured for a CSP on the Data Sources page. You can check the status, edit, delete, enable, or disable instances, and initiate discovery scan.

1. Select Settings → Data Sources.
2. Find the cloud instance by clicking on the CSP name or using the Search field.
3. In the row for the cloud instance, click View Details. The Cloud Instances page is displayed, filtered by the CSP you selected.
4. In the Cloud Instances page, you can filter the results by any heading and value.
5. Click on an instance name to open the details pane for that instance.
6. You can perform the following actions on each cloud instance:

Action	Instructions
Discover Now	To initiate a discovery scan, in the row for the cloud instance, right-click and select Discover Now. Alternatively, in the details pane, click the more options icon and select Discover Now.

Action	Instructions
Enable/Disable	In the row for the cloud instance, right-click and select Enable or Disable. Alternatively, in the details pane, click the more options icon and select Enable or Disable.
Delete	In the row for the cloud instance, right-click and select Delete. Alternatively, in the details pane, click the more options icon and select Delete.
Create a new instance	Click New Instance and select the type of CSP of which you want to create a new instance. Follow the onboarding wizard to define its settings.
Edit configuration	In the row for the cloud instance, right-click and select Configuration. Alternatively, in the details pane, click the edit button. Follow the onboarding wizard to edit the cloud instance's settings. You must execute the updated template in the CSP environment for the configuration changes to be applied.

1.4.4 | Update cloud permissions after Cortex release updates

Abstract

Manage permission updates for your cloud instances following new feature releases or bug fixes.

This topic provides guidance on how to manage permission updates for your cloud instances following new feature releases or bug fixes. It outlines how users are notified of required permission changes and provides step-by-step instructions for granting necessary permissions to ensure continued functionality and security.

> **PREREQUISITE:**
>
> - Ensure that the user account used to modify permissions has the necessary privileges within both the Cortex platform and your cloud environment, for example, AWS or Azure.
> - You received a notification regarding a new version available that requires permission updates, or viewed a Needs Update status in the Data Sources page.

Procedure

1. Navigate to the Data Sources page.
2. Do the following to identify instances requiring updates:
    a. For the relevant instance, locate the Update Status column.
    b. Filter or sort by this column to quickly identify instances marked as Needs Update. The message on the page indicates the number of instances that need updating.

    > **NOTE:**
    >
    > Instances requiring updates will not change their connection status, for example, Connected, Warning, Error, Disabled, due to the pending permission update.
3. Do the following to access the connector's permissions section:
    a. Click the name of the specific cloud connector instance that requires permission updates. The connector's detailed view appears.
    b. Within the connector's detailed view, locate and select the permissions section.
4. Review missing permissions. In the permissions section, the missing permission names or changes in permission scope is indicated.
5. Follow the on-screen instructions to grant the required permissions, or refer to the specific permission names or scopes provided.
6. After making the necessary permission adjustments, click Save or Apply Changes within the connector's configuration.
7. Return to the Data Sources page and verify that the updated status of the instance shows as up-to-date, or the update is in progress.

8. Monitor the instance's health and functionality to confirm the changes have taken effect and the connector is operating as expected.

   If you encounter issues during the permission update process, check the generated health alerts for more specific details.

1.4.5 | Pending cloud instances

In Cortex XSIAM, a pending cloud instance refers to a cloud instance created after Cortex XSIAM generates an authentication template, but before that template has been fully executed within the Cloud Service Provider (CSP) environment.

A pending cloud instance is created each time you complete the onboarding wizard for a new CSP and click Save. You can view all cloud instances, including those in a pending state, by navigating to Cloud Instances. Ensure you remove any default filters that might exclude instances with a "pending" status.

A single pending instance can be leveraged to create multiple cloud instances, all sharing the same configurations defined during the cloud onboarding process. Pending instances are automatically deleted after 30 days.

Manage pending cloud instances

There are some actions that can be performed specifically on cloud instances with a status of "pending".

Action	Instructions
Manually connect an instance	After the authentication template has been executed in the CSP, you can manually connect the Cortex XSIAM cloud instance to the CSP by right-clicking the pending cloud instance and selecting Manually connect an instance. For more about this process, see Manually connect a cloud instance.
View Details	To review the configuration settings defined in the onboarding wizard for a pending instance, right-click the instance and select View Details. This is helps you distinguish between pending instances when you want to create a new cloud instance from an existing pending instance or when you want to manually connect an instance.
Re-download Connection Template	The authentication template that you download from the onboarding wizard is valid for seven days from when it was downloaded. If you want to create a new cloud instance from a pending instance after the authentication template has expired, you can right-click the pending instance and select Re-download Connection Template. You must then execute the template in the CSP.
Delete	To delete a pending instance, right-click the pending instance and select Delete.

1.4.6 | Troubleshoot errors on cloud instances

Abstract

You can troubleshoot errors on cloud instances by drilling down on an instance from the Data Sources page.

To help you to troubleshoot errors on a cloud instance, Cortex XSIAM provides the following visibility and drilldown options:

- Overall status of an instance that indicates the health of your instance.
- A breakdown of the security capabilities enabled on an instance, detailing the status of each capability along with any open errors or issues.
- Additional XQL drill down options to query the history of error and recovery events for each security capability.

How to troubleshoot errors on a cloud instance

1. Go to Settings → Data Sources.

   Under Cloud Service Provider, review the status of the instances that were onboarded for the service provider. If the status shows Warning or Error, hover over the service provider and click View Details.

2. On the Cloud Instances page review the list of instances that were onboarded and their overall status. The status is displayed as follows:

    - Connected: The connector is enabled and has no issues.
    - Warning: The connector is enabled and has minor issues. For example, some accounts or capabilities are in warning or error status.
    - Error: The connector is enabled and has substantial errors. For example, an authentication failure, an outpost failure, major permissions issues, or (for organization level accounts) the majority of the accounts in the instance are in error status.
    - Disabled: The connector is disabled.

3. To understand why an instance is showing a Warning or Error status, click on the instance name.

    The cloud instance panel provides a breakdown of the security capabilities and the accounts onboarded on the instance. Review the information in the following sections:

Section	Context
Header	Displays the overall status of the instance and the following information about the account, as specified during onboarding: • Scope of the instance: The number of accounts onboarded on the instance and their status. See the Accounts section for more information about the individual accounts and the type of account (single account or organization). • Scan mode: Cloud Scan or Outpost. For accounts using Outpost, information is displayed about the status of the Outpost account and the account ID. • Resource Tags: Tags defined during onboarding.
Security Capabilities	Displays a breakdown of the security capabilities enabled on the instance and their individual statuses. Click on any item that shows a warning or error status to see the open errors and issues that contributed to the status: • Errors are factual objects that are automatically created when problems occur, and provide insight into the current status of the capability. For example, if a permission is missing, an error is displayed. Browse and filter the errors to better understand and resolve the problem. • Issues are actionable objects that are triggered when detected problems exceed defined thresholds. Issues are manageable, trackable, and provide remediation suggestions and automations. The issues displayed in the panel are open issues that are specifically related to the selected connector with the selected capability in the observed scope (single account or organization). Click an issue to start investigating it.
Accounts	Lists the accounts that are onboarded on the instance and their individual status. If multiple accounts are onboarded on the instance, click on each account to filter the page information by account, and drill-down to the security capability statuses for each account.

4. If the instance shows an Outpost error, go to the All Outposts page and find the outpost account that is being used by this instance. Right click the Outpost account to view the open errors and issues for the account.
5. If the account shows Permission errors, use the side panel to check which permissions are missing. You can also Edit the instance to redeploy the cloud setup template, which should normally resolve the error.
6. Further investigate errors by running XQL queries on the `cloud_health_auditing` dataset.

    This dataset records error and recovery events for the security capabilities in cloud instances. By querying this dataset you can see information about when the error started, the prevalence of the error, and whether there is a recurrency pattern. See the specific fields descriptions and query examples for each security capability.

    > **NOTE:**
    >
    > Errors related to collection of audit logs in the cloud instance are recorded in the `collection_auditing` dataset. For more information, see Audit logs fields and query examples.

7. Set up correlation rules to trigger issues when errors occur in cloud security capabilities. See the following examples.

Outpost fields and query examples

You can review Outpost entries in the `cloud_health_auditing` dataset to see Outpost activity over time, or to search for errors on specific accounts. Outpost entries are added to the dataset as follows:

- An error occurred on an Outpost account that disabled or prevented an operation. This is audited as Error.
- An exceptional condition occurred on an Outpost account that might cause problems if not resolved. This is audited as Warning.
- The Outpost account returns to normal function. This is audited as Informational.

The following table describes the fields for Outpost entries:

Field	Description
Account	Cloud account ID of the Outpost
Name	Category of the error, or a brief description of the event
Resource ID	Outpost ID
Capability	Outpost
Region	Region where the event occurred, or All regions.
Classification	Type of entry (Error, Warning, or Informational)
Message	Description of the error or Connected for informational entries.
Error	Details about the error. For informational entries this is blank.

Example 155. Examples of Outpost queries

- Identify Outpost errors on all Outpost accounts in the eu-west-3 region:

```
dataset = cloud_health_auditing | filter capability = "Outpost" and classification = "Error" and region = "eu-west-3"
```

- See all entries (error, warning, and recovery) for Outpost_1 on cloud account Account_A:

```
dataset = cloud_health_auditing | filter capability = "Outpost" and resource_id = “Outpost_1” and account = "Account_A"
```

Permissions fields and query examples

You can review Permissions entries in the `cloud_health_auditing` dataset to see Permissions activity over time, or to search for errors on specific accounts. Permissions entries are added to the dataset as follows:

- A permission problem was found. This is audited as Error.
- An exceptional condition occurred that might cause problems if not resolved. This is audited as Warning.
- A permission problem is resolved. This is audited as Informational.

The following table describes the fields for Permissions entries:

Field	Description
Account	Name of the account where the event occurred, or All accounts.
Connector	Name of the connector where the event occurred

Field	Description
Name	Permission name
Capability	Permissions
Classification	Type of entry (Error, Warning, or Informational)
Message	Description of the error or Granted for informational entries.

Discovery engine fields and query examples

You can review Discovery engine entries in the `cloud_health_auditing` dataset to see Discovery activity over time, or to search for errors on specific accounts. Discovery entries are added to the dataset as follows:

- An API exec problem is found. This is audited as Error.
- An exceptional condition occurred that might cause problems if not resolved. This is audited as Warning.
- An API exec problem is resolved. This is audited as Informational.

The following table describes the fields for Discovery engine entries:

Field	Description
Account	Name of the account where the event occurred, or All accounts
Connector	Name of the connector where the event occurred
Name	Asset name
Capability	Discovery
Region	Region where the event occurred, or All regions.
Classification	Type of entry (Error, Warning, or Informational)
Message	Description of the error or Connected for informational entries.

Example 156. Examples of Discovery engine queries

- Identify API exec errors on the Discovery engine for all accounts on the AWS_1 connector:

```
dataset = cloud_health_auditing | filter capability = "Discovery" and connector = "AWS_1" and classification = “Error”
```

- See all Discovery engine activity on connector AWS_1 for Account_ A in the af-south-1 region:

```
dataset = cloud_health_auditing | filter capability = "Discovery" and connector = "AWS_1" and account = "accountA" and region = "af-south-1"
```

Agentless Disk Scanning (ADS) fields and query examples

You can review ADS entries in the `cloud_health_auditing` dataset to see ADS activity over time, or to search for errors on specific accounts. ADS entries are added to the dataset as follows:

- ADS failed to scan an asset. This is audited as Failed.
- ADS successfully scanned an asset. This is audited as Scanned.
- The asset or host is not supported by ADS. This is audited as Unsupported.
- The asset or Host was excluded from the scan. This is audited as Excluded.

Field	Description
Account	Name of the account to which the asset belongs
Connector	ID of the connector
Name	Name of the asset
Resource ID	Asset ID
Capability	ADS
Region	Region where the asset is located
Classification	Type of entry (Failed, Unsupported, Excluded, Scanned)
Message	Description of the error, or Connected for informational entries.
Error	Details about the error. For informational entries this is blank.
Type	Type of asset that was scanned
Scope	Scope of the asset (Asset, Region, or Account)

Example 157. Examples of ADS queries

- Identify failed ADS scans on connector "a8df43e848dd42778ae7efd5a706a0fc" for EC2 assets at the asset scope level, filtered by region (northamerica-northeast2-a):

```
dataset = cloud_health_auditing | filter capability = "ADS" and classification = "failed" and connector = “a8df43e848dd42778ae7efd5a706a0fc” and type = "EC2_INSTANCE" and scope = "Asset" and region = "northamerica-northeast2-a"
```

- See all ADS scans (failed and successful) on connector "a8df43e848dd42778ae7efd5a706a0fc" for EC2 assets belonging to Account_A:

```
dataset = cloud_health_auditing | filter capability = "ADS" and connector = “a8df43e848dd42778ae7efd5a706a0fc” and type = "EC2" and account = “Account_A”
```

Data Security Scanning (DSPM) fields and query examples

You can review DSPM entries in the `cloud_health_auditing` dataset to see DSPM activity over time, or to search for errors on specific accounts. DSPM entries are added to the dataset as follows:

- DSPM failed to scan an asset. This is audited as Failed.
- DSPM successfully scanned an asset. This is audited as Success.

The following table describes the fields for DSPM entries:

Field	Description
Account	Name of the account to which the asset belongs
Connector	Name of the connector where the event occurred
Name	Name of the asset
Resource ID	Asset ID
Capability	DSPM
Region	Region where the asset is located
Classification	Type of entry (Failed or Success)
Message	Description of the error, or Connected for informational entries.
Error	Details about the error. For informational entries this is blank.
Type	Type of asset that was scanned
Scope	Scope of the asset (Asset, Region, or Account)

Example 158. Examples of DSPM queries

- Identify failed DSPM scans on the AWS_1 connector for S3 asset types, filtered by region (ap-east-1):

```
dataset = cloud_health_auditing | filter capability = "DSPM" and classification = “Error” and connector = “AWS_1” and type = "S3_BUCKET" and region = "ap-east-1"
```

- See all DSPM scans (failed and successful) on the AWS_1 connector, for all scanned assets on Account_A:

```
dataset = cloud_health_auditing | filter capability = "DSPM" and account = "Account_A" and connector = “AWS_1”
```

Registry scanning fields and query examples

You can review Registry scanning entries in the `cloud_health_auditing` dataset to see Registry scanning activity over time, or to search for errors on specific accounts. Registry scanning entries are added to the dataset as follows:

- The Registry scanner failed to scan an asset. This is audited as Failed.
- The Registry scanner successfully scanned an asset. This is audited as Scanned.

The following table describes the fields for Registry scanning entries:

Field	Description
Account	Name of the account to which the asset belongs
Connector	Name of the connector where the event occurred

Field	Description
Resource ID	Asset ID
Capability	Registry
Classification	Type of entry (Scanned or Failed)
Error	Details about the error. For informational entries this is blank
Scope	Scope of the asset (Asset or Account)

Example 159. Examples of Registry scanning queries

- Identify failed scans on connector GCP_1:

```
dataset = cloud_health_auditing | filter capability = "Registry" and classification = “error” and connector = “GCP_1”
```

- Review all registry scans (failed and successful) on connector GCP_1 for asset Asset_A:

```
dataset = cloud_health_auditing | filter capability = "Registry" and connector = “GCP_1” and ressource_id = "Asset_A"
```

Audit logs fields and query example

You can review Audit logs entries in the `collection_auditing` dataset. Querying this dataset can help you see the connectivity changes of an instance over time, the escalation or recovery of the connectivity status, and the error, warning, and informational messages related to status changes. For more information about this dataset, see Verify collector connectivity.

The following table describes the fields for Audit logs entries:

Field	Description
Instance	Instance name
Log type	Type of logs affected
Classification	Type of entry (Error, Warning, or Informational)
Collector type	Type of the collector
Description	Description of the error, or Connected for informational entries.

Example 160. Audit logs query example

Identify disruptions (errors) in audit log collection on connector AWS_1:

```
dataset = collection_auditing | filter instance = “AWS_1” and log_type = "Audit Logs" and classification = “Error”
```

Correlation rule examples

The following examples show how to set up correlation rules to trigger Health Collection issues when errors occur on a specific security capability.

Example rule for DSPM errors

In this example, a correlation rule will trigger a Health Collection issue if a DSPM scan fails on an AWS_S3 asset on the AWS_1 connector.

Example XQL:

```
dataset = cloud_health_auditing | filter capability = "DSPM" and classification = “Error” and type = "AWS_S3" and scope = "Asset" and connector = “AWS_1”
```

Additional fields to specify in the correlation rule:

Field	Value
Time Schedule	Hourly
Query time frame	1 Hour
Issue Suppression	Select Enable issue suppression.
Action	Select Generate Issue.
Issue Domain	Health
Severity	Medium
Category	Collection

Example rule for Outpost errors

In this example, a correlation rule will trigger a Health Collection issue if an error is recorded on account Outpost_A in the us-east-1 region.

Example XQL:

```
dataset = cloud_health_auditing | filter capability = "Outpost" and account = "Outpost_A" and region = "eu-west-3" and classification = "Error"
```

Additional fields to specify in the correlation rule:

Field	Value
Time Schedule	Hourly
Query time frame	1 Hour
Issue Suppression	Select Enable issue suppression.
Action	Select Generate Issue.
Issue Domain	Health
Severity	Medium
Category	Collection

1.4.7 | Manage Kubernetes Connector instances

Abstract

You can manage the Kubernetes Connector instances on the Data Sources page. You can check the status, edit or delete Kubernetes Connector instances.

1. Select Settings → Data Sources.
2. Find the Kubernetes instance by clicking on the Kubernetes name or using the Search field.
3. In the row for the Kubernetes instance, click View Details. The Kubernetes Connectors page is displayed with all deployed Kubernetes Connectors. To view all Kubernetes clusters, including ones that are not yet deployed, go to the Kubernetes Connectivity Management page.
4. In the Kubernetes Connectors page, click on a cluster name to open the details pane for that instance.
5. You can perform the following actions on each Kubernetes Connector instance:

Action	Instructions
Open Cluster Details	In the details pane, click the more options icon and select Open Cluster Details. The Asset Card for that Kubernetes cluster is displayed.
Edit Connector	In the row for the Kubernetes instance, right-click and select Edit. Alternatively, in the details pane, click the more options icon and select Edit Connector. In Edit Kubernetes Connector, enter a name for the installer. You can edit the namespace for the connector, the scan cadence, and the version of the connector you want to install. You must execute the updated template in the Kubernetes environment for the configuration changes to be applied.
Delete Connector	In the row for the Kubernetes instance, right-click and select Delete. Alternatively, in the details pane, click the more options icon and select Delete Connector. To remove the connector, you must manually run Kubernetes commands to delete the resources in the Kubernetes environment. The commands are listed here.

Kubernetes Connectivity Management

Navigate to Settings → Data Sources and find the Kubernetes instances by clicking on the Kubernetes name or using the Search field. In the Kubernetes Connectors page, click Kubernetes Connectivity Management to view all detected Kubernetes clusters. Here, you can check if a cluster is connected, view the status, and see the connector version. When a new version of the Kubernetes Connector is available, you can update it here.

1.5 | Palo Alto Networks integrations

Abstract

Cortex XSIAM supports data ingestion from other Palo Alto Networks products.

> **LICENSE TYPE:**
>
> Data collection may require an add-on.

Cortex XSIAM supports streaming data directly from Prisma Access accounts, Next-Generation Firewalls (NGFW), and Panorama devices to your Cortex XSIAM tenants using the Strata Logging Service.

New tenants (and tenants upgraded from XDR to XSIAM) will work with the new direct integration of Next-Generation Firewall and Panorama into Cortex. For such tenants, there’s no option to use the Strata Logging Service integration.

For tenants where a Strata Logging Service license exists, the configured integrations, such as Next-Generation Firewall and Prisma Access, can be migrated to Cortex XSIAM in either of the following ways before the license expires:

- More than two weeks before the license for existing integrations with Strata Logging Service expires, manually migrate the integrations, using the corresponding Migrate Devices buttons on the Data Sources page. Make sure you select all your devices to connect directly to Cortex XSIAM.
- Two weeks prior to the end of your Strata Logging Service license, Cortex XSIAM will automatically migrate your integrations to your Strata Logging Service.

  > **NOTE:**
  >
  > Roll-back of Strata Logging Service integration migration is not supported.

1.5.1 | About Palo Alto Networks integrations

Abstract

Stream data directly from other Palo Alto Networks products to Cortex XSIAM.

Cortex XSIAM supports streaming data directly from Prisma Access accounts, Next-Generation Firewalls (NGFW), and Panorama devices to your Cortex XSIAM tenants using the Strata Logging Service.

Ensure you have deployed Panorama and NGFW, and hold Super User permissions to your Customer Support Account (CSP).

After your tenant has been activated, navigate to the Data Sources page to configure your integrations. All devices and accounts allocated to your CSP accounts are available to integrate.

> **NOTE:**
>
> For Palo Alto Networks Integrations there is an option to turn on or off the collection of URL and File log types. For more information, see Collecting URL and File log types.

New tenants (and tenants upgraded from XDR to XSIAM) will work with the new direct integration of Next-Generation Firewall and Panorama into Cortex. For such tenants, there's no option to use the Strata Logging Service integration.

For tenants where a Strata Logging Service license exists, the configured integrations, such as Next-Generation Firewall and Prisma Access, can be migrated to Cortex XSIAM in either of the following ways before the license expires:

- More than two weeks before the license for existing integrations with Strata Logging Service expires, manually migrate the integrations, using the corresponding Migrate Devices buttons on the Data Sources page. Make sure you select all your devices to connect directly to Cortex XSIAM.
- Two weeks prior to the end of your Strata Logging Service license, Cortex XSIAM will automatically migrate your integrations to your Strata Logging Service.

  > **NOTE:**
  >
  > Roll-back of Strata Logging Service integration migration is not supported.

1.5.2 | Ingest data from Next-Generation Firewall

Abstract

Learn how to ingest detection data from Next-Generation Firewall and Panorama.

You can forward firewall data from your Next-Generation Firewall (NGFW) and Panorama devices to Cortex XSIAM.

Collection of firewall data from multiple accounts is supported. Super User permissions on both the Cortex XSIAM tenant accounts and the NGFW or Panorama accounts are required for this use case.

New tenants (and tenants upgraded from XDR to XSIAM) will work with the new direct integration of Next-Generation Firewall and Panorama into Cortex. For such tenants, there's no option to use the Strata Logging Service integration.

For tenants where a Strata Logging Service license exists, the configured integrations, such as Next-Generation Firewall and Prisma Access, can be migrated to Cortex XSIAM in either of the following ways before the license expires:

- More than two weeks before the license for existing integrations with Strata Logging Service expires, manually migrate the integrations, using the corresponding Migrate Devices buttons on the Data Sources page. Make sure you select all your devices to connect directly to Cortex XSIAM.
- Two weeks prior to the end of your Strata Logging Service license, Cortex XSIAM will automatically migrate your integrations to your Strata Logging Service.

  > **NOTE:**
  >
  > Roll-back of Strata Logging Service integration migration is not supported.

> **PREREQUISITE:**
>
> Ensure that you have completed the following on the NGFW or Panorama side:
>
> - For Panorama only, ensure that the Panorama Cloud Services plugin is installed.
> - Enable log forwarding profiles on firewall rules.
>
> On the Cortex XSIAM side, ensure that you have user role permissions for Data Collection > Data Sources.
>
> Configuration of data ingestion from multiple accounts requires Super User permissions on both the Cortex XSIAM tenant and on the device accounts.

> **NOTE:**
>
> If your firewalls are located in a different region, or bandwidth issues are encountered due to large log size, you can ingest NGFW logs in CEF format, using the Syslog collector. However, the Syslog solution is not as powerful nor as comprehensive as this data collector, and should only be used when this data collector cannot be used. For more information, see Ingest Next-Generation Firewall logs using the Syslog collector.

> **NOTE:**
>
> In the following procedure, general information is provided for NGFW and Panorama. For detailed instructions, consult the documentation for your specific devices and Panorama version.

Set up detection data ingestion

1. In the user interface for setting up firewalls, for Strata Logging Service/Cloud Logging, enable the following options directly, or using device templates.

   (For example, go to Device → Setup → Management → Cloud Logging section)

   a. Select Enable Strata Logging Service.

   b. Select Enable Enhanced Application Logging.

   c. (Optional, depending on your organization's requirements) Select Enable Duplicate Logging (Cloud and On-Premise).

2. Depending on your PAN-OS or Panorama version, generate either a certificate or PSK.

   For PAN-OS and Panorama versions 10.1 and later, each firewall requires a separate certificate. Certificates need to be requested through the Customer Support portal. To sign in to the portal, click here. For PAN-OS and Panorama versions 10.0 and earlier, you are only required to generate one global PSK for all the firewall devices.

   > **NOTE:**
   >
   > Cortex XSIAM does not validate your firewall credentials, you must ensure the certificates or PSK details have been updated in your firewalls in order for data to stream.

3. Onboard the certificates.
4. Define a Log Forwarding profile.
5. Map the Log Forwarding profile to a Security Policy Rule.
6. Verify that the connection between the firewalls and Strata Logging Service is valid.
7. Push the configuration changes to the firewalls.
8. In Cortex XSIAM, select Settings → Data Sources.
9. On the Data Sources page, click Add Data Source, search for and select NGFW, and click Connect.
10. Select Add NGFW Device or Add Panorama Device, and then do one of the following:

    - For devices in your account, select one or more devices from Select FW/Panorama devices.
    - To include devices from other accounts, select Select devices from other accounts, and then select one or more FW or Panorama devices from other accounts. For cross-account connections, you must have Super User permissions on the Cortex tenant account and the device account.

    Devices already connected are listed at the end. A device may be connected via Strata Logging Service, or via Cortex XSIAM. Rectify any streaming issues that may arise by checking configurations for the relevant connection type (Strata Logging Service or Cortex XSIAM).

11. To complete the onboarding process of your devices, on the Next Steps to Connect Your Devices page, expand the relevant device version, and follow the corresponding instructions.
12. Click Connect to establish the instance.

    Connection is established regardless of the firewall credential status and can take up to several minutes, select Sync now to refresh your instances.

13. Validate that your data is streaming. It might be necessary to create traffic before you verify data streaming.

    To ensure the data is streaming into your tenant:

    - In your NGFW Standalone Firewall Devices, track the Last communication timestamp.
    - Run XQL Query: dataset = panw_ngfw_system_raw| filter log_source_id = "[NGFW device SN]"

14. (Optional) Manage your Instance.

    After you create the NGFW instance, on the Data Sources page, expand the NGFW to track the status of your Standalone Firewall Devices and Panorama Devices.

    Select the ellipses to Request Certificate, if required, or Delete the instance.

> **NOTE:**
>
> It might take an hour or longer after connecting the firewall in Cortex XSIAM until you start seeing notifications that the certificate has been approved, and that the logging service license has appeared on the firewall.

When Cortex XSIAM begins receiving detection data, the console begins stitching logs with other Palo Alto Network-generated logs to form stories. Use the XQL Search dataset `panw_ngfw_*_raw` to query your data, where the following logs are supported:

- Authentication Logs: panw_ngfw_auth_raw
- File Data Logs: panw_ngfw_filedata_raw
- Global Protect Logs: panw_ngfw_globalprotect_raw
- Hipmatch Logs: panw_ngfw_hipmatch_raw*
- System Logs: panw_ngfw_system_raw
- Threat Logs: panw_ngfw_threat_raw*
- Traffic Logs: panw_ngfw_traffic_raw*
- URL Logs: panw_ngfw_url_raw*
- User ID Logs: panw_ngfw_userid_raw
- Configuration Logs: panw_ngfw_config_raw
- Tunnel Logs: panw_ngfw_tunnel_raw

*These datasets use the query field names as described in the Cortex schema documentation.

For stitched raw data, you can query the `xdr_data` dataset or use any preset designated for stitched data, such as `network_story`. For query examples, refer to the in-app XQL Library. When relevant, Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC only) from Strata Logging Service detection data. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

> **NOTE:**
>
> IOC and BIOC issues are applicable on stitched data only, and are not available on raw data.

> **TIP:**
>
> You can see an overview of ingestion status for all log types, and a breakdown of each log type and its daily consumption quota on the NGFW Ingestion Dashboard.

1.5.2.1 | Ingest Next-Generation Firewall logs using the Syslog collector

Abstract

Use the Syslog collector to ingest NGFW logs in CEF format. This method is useful when your firewalls are located in a different region, or bandwidth issues are encountered due to large log size.

Use the Syslog collector to ingest Next-Generation Firewall (NGFW) logs in CEF format. This method is useful when your firewalls are located in a different region, or bandwidth issues are encountered due to large log size. When possible, we recommend that you ingest NGFW logs using the dedicated Next-Generation Firewall data collector instead of the Syslog collector.

> **NOTE:**
>
> In the following procedure, general information is provided for NGFW and Panorama. For detailed instructions, consult the documentation for your specific devices and Panorama version, to ensure that you have configured log forwarding correctly for all the log types that you would like to forward to Cortex XSIAM. The following steps only cover configuration of the custom log schema (CEF) for a given syslog server. They do not replace the administrator guide's configuration coverage of log forwarding.

Configure the firewall/Panorama for log forwarding to Cortex XSIAM

1. To configure the device to include its IP address in the header of Syslog messages, select Panorama/Device → Setup → Management, click the Edit icon in the Logging and Reporting Settings section, and navigate to the Log Export and Reporting tab.
2. From the Syslog HOSTNAME Format menu, select ipv4-address or ipv6-address, and click OK.
3. Select Device → Server Profiles → Syslog, and click Add.
4. Enter a server profile Name and Location (Location refers to a virtual system, if the device is enabled for virtual systems).
5. On the Servers tab of the Syslog Server Profiles window, click Add, and enter the following information for the Syslog server:
   - Name
   - Syslog Server (IP address)
   - Transport, Port (default 514 for UDP)
   - Facility (default LOG_USER)
6. Select the Custom Log Format tab and click configure the log formats as follows:

   > **NOTE:**

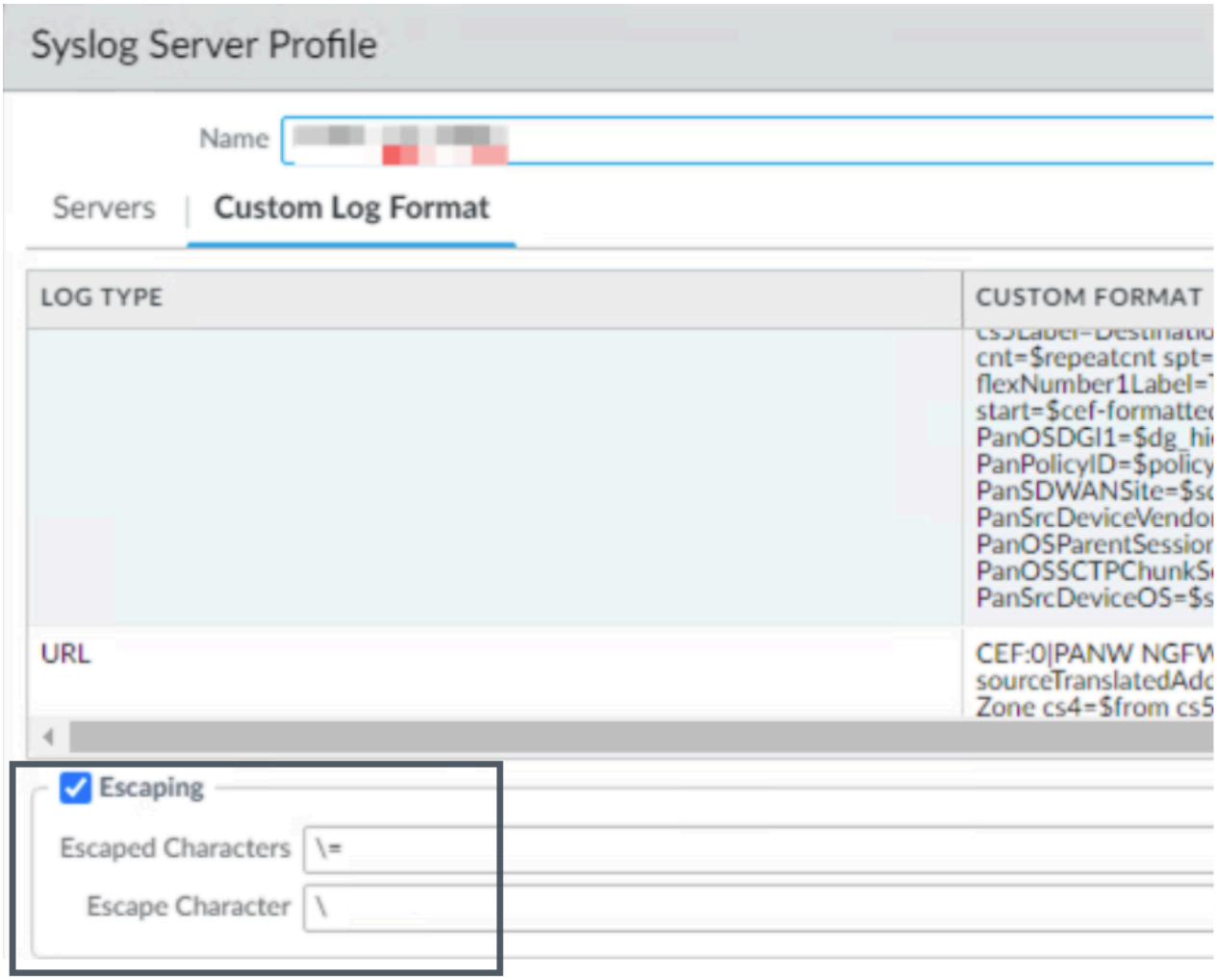
To avoid the possible effects of line formatting, do not copy/paste the message formats directly into the PAN-OS web interface. Instead, paste into a text editor, remove any carriage return or line feed characters, and then copy and paste into the web interface.
NOTE:

From version 10.0 and later, the log format documented for log types (Traffic, Threat, and URL) exceeds the maximum supported 2048 characters in the Custom Log Format tab on the firewall and Panorama. Select the CEF keys and values to limit the number of characters to 2048, as per your requirements.

Log Type	Custom Format
Traffic	CEF:0|PANW|NGFW_CEF|$sender_sw_version\|$subtype|$type\|1\| __firewall_type=firewall.traffic __timestamp=$start __tz=$high_res_timestamp log_type=$type subtype=$subtype log_time=$cef-formatted-receive_time time_generated=$cef-formatted-time_generated log_source_id=$serial log_source_name=$device_name sequence_no=$seqno source_ip=$src dest_ip=$dst source_port=$sport dest_port=$dport nat_source=$natsrc nat_dest=$natdst nat_source_port=$natsport nat_dest_port=$natdport protocol=$proto action=$action source_user=$srcuser dest_user=$dstuser xff_ip=$xff_ip app=$app app_category=$category_of_app app_sub_category=$subcategory_of_app rule_matched=$rule rule_matched_uuid=$rule_uuid severity=1 vsys=$vsys vsys_name=$vsys_name from_zone=$from to_zone=$to inbound_if=$inbound_if outbound_if=$outbound_if session_id=$sessionid source_device_category=$src_category source_device_profile=$src_profile source_device_model=$src_model source_device_vendor=$src_vendor source_device_osfamily=$src_osfamily source_device_osversion=$src_osversion source_device_mac=$src_mac dest_device_category=$dst_category dest_device_profile=$dst_profile dest_device_model=$dst_model dest_device_vendor=$dst_vendor dest_device_osfamily=$dst_osfamily dest_device_osversion=$dst_osversion dest_device_mac=$dst_mac bytes_sent=$bytes_sent bytes_received=$bytes_received packets_received=$pkts_received packets_sent=$pkts_sent total_time_elapsed=$elapsed session_end_reason=$session_end_reason url_category=$category
Threat	CEF:0|PANW|NGFW_CEF|$sender_sw_version\|$threatid|$type\|$number-of-severity| __firewall_type=firewall.threat __timestamp=$cef-formatted-time_generated __tz=$high_res_timestamp log_type=$type subtype=$subtype log_time=$cef-formatted-receive_time time_generated=$cef-formatted-time_generated log_source_id=$serial log_source_name=$device_name sequence_no=$seqno source_ip=$src dest_ip=$dst source_port=$sport dest_port=$dport nat_source=$natsrc nat_dest=$natdst nat_source_port=$natsport nat_dest_port=$natdport protocol=$proto action=$action source_user=$srcuser dest_user=$dstuser xff=$xff xff_ip=$xff_ip app=$app app_category=$category_of_app app_sub_category=$subcategory_of_app rule_matched=$rule rule_matched_uuid=$rule_uuid severity=$number-of-severity vsys=$vsys vsys_name=$vsys_name from_zone=$from to_zone=$to inbound_if=$inbound_if outbound_if=$outbound_if session_id=$sessionid source_device_category=$src_category source_device_profile=$src_profile source_device_model=$src_model source_device_vendor=$src_vendor source_device_osfamily=$src_osfamily source_device_osversion=$src_osversion source_device_mac=$src_mac dest_device_category=$dst_category dest_device_profile=$dst_profile dest_device_model=$dst_model dest_device_vendor=$dst_vendor dest_device_osfamily=$dst_osfamily dest_device_osversion=$dst_osversion dest_device_mac=$dst_mac misc=$misc threat_id=$threatid threat_name=$threat_name threat_category=$thr_category direction=$direction user_agent=$user_agent
URL	CEF:0|PANW|NGFW_CEF|$sender_sw_version\|$subtype|$type\|$number-of-severity| __firewall_type=firewall.url __timestamp=$cef-formatted-time_generated __tz=$high_res_timestamp log_type=$type subtype=$subtype log_time=$cef-formatted-receive_time time_generated=$cef-formatted-time_generated log_source_id=$serial log_source_name=$device_name sequence_no=$seqno source_ip=$src dest_ip=$dst source_port=$sport dest_port=$dport nat_source=$natsrc nat_dest=$natdst nat_source_port=$natsport nat_dest_port=$natdport protocol=$proto action=$action source_user=$srcuser dest_user=$dstuser xff=$xff xff_ip=$xff_ip app=$app app_category=$category_of_app app_sub_category=$subcategory_of_app rule_matched=$rule rule_matched_uuid=$rule_uuid severity=$number-of-severity vsys=$vsys vsys_name=$vsys_name from_zone=$from to_zone=$to inbound_if=$inbound_if outbound_if=$outbound_if session_id=$sessionid source_device_category=$src_category source_device_profile=$src_profile source_device_model=$src_model source_device_vendor=$src_vendor source_device_osfamily=$src_osfamily source_device_osversion=$src_osversion source_device_mac=$src_mac dest_device_category=$dst_category dest_device_profile=$dst_profile dest_device_model=$dst_model dest_device_vendor=$dst_vendor dest_device_osfamily=$dst_osfamily dest_device_osversion=$dst_osversion dest_device_mac=$dst_mac uri=$misc threat_id=$threatid threat_name=$threat_name threat_category=$thr_category direction=$direction user_agent=$user_agent url_category=$category url_category_list=$url_category_list content_type=$contenttype http_method=$http_method http_headers=$http_headers http2_connection=$http2_connection referer=$referer pcap_id=$pcap_id

Log Type	Custom Format
File Data	CEF:0|PANW|NGFW_CEF|$sender_sw_version\|$threatid|$type\|$number-of-severity| __firewall_type=firewall.filedata __timestamp=$cef-formatted-time_generated __tz=$high_res_timestamp log_type=$type subtype=$subtype log_time=$cef-formatted-receive_time time_generated=$cef-formatted-time_generated log_source_id=$serial log_source_name=$device_name sequence_no=$seqno source_ip=$src dest_ip=$dst source_port=$sport dest_port=$dport nat_source=$natsrc nat_dest=$natdst nat_source_port=$natsport nat_dest_port=$natdport protocol=$proto action=$action source_user=$srcuser dest_user=$dstuser xff=$xff xff_ip=$xff_ip app=$app app_category=$category_of_app app_sub_category=$subcategory_of_app rule_matched=$rule rule_matched_uuid=$rule_uuid severity=$number-of-severity vsys=$vsys vsys_name=$vsys_name from_zone=$from to_zone=$to inbound_if=$inbound_if outbound_if=$outbound_if session_id=$sessionid source_device_category=$src_category source_device_profile=$src_profile source_device_model=$src_model source_device_vendor=$src_vendor source_device_osfamily=$src_osfamily source_device_osversion=$src_osversion source_device_mac=$src_mac dest_device_category=$dst_category dest_device_profile=$dst_profile dest_device_model=$dst_model dest_device_vendor=$dst_vendor dest_device_osfamily=$dst_osfamily dest_device_osversion=$dst_osversion dest_device_mac=$dst_mac misc=$misc threat_id=$threatid threat_name=$threat_name threat_category=$thr_category direction=$direction user_agent=$user_agent file_url=$file_url filedigest=$filedigest filetype=$filetype pcap_id=$pcap_id

7. Configure Escaping characters as follows:
    - Escaped Characters: \=
    - Escape Character: \



Configure Syslog collection

Set up a Syslog collector for the logs, as explained in Activate Syslog Collector. In Task 4, ensure that you set Format to CEF.

1.5.3 | Ingest data from Prisma Access

Abstract

Learn how to ingest detection data from Prisma Access.

You can forward data from Prisma Access to Cortex XSIAM. When your Cortex XSIAM tenant begins receiving detection data, it begins stitching logs with other Palo Alto Networks-generated logs to form stories. Use the XQL Search to query the data.

Collection of data from multiple accounts is supported. Super User permissions on both the Cortex XSIAM tenant accounts and the Prisma Access accounts are required for this use case.

New tenants (and tenants upgraded from XDR to XSIAM) will work with the new direct integration of Next-Generation Firewall and Panorama into Cortex. For such tenants, there's no option to use the Strata Logging Service integration.

For tenants where a Strata Logging Service license exists, the configured integrations, such as Next-Generation Firewall and Prisma Access, can be migrated to Cortex XSIAM in either of the following ways before the license expires:

- More than two weeks before the license for existing integrations with Strata Logging Service expires, manually migrate the integrations, using the corresponding Migrate Devices buttons on the Data Sources page. Make sure you select all your devices to connect directly to Cortex XSIAM.
- Two weeks prior to the end of your Strata Logging Service license, Cortex XSIAM will automatically migrate your integrations to your Strata Logging Service.

  > **NOTE:**
  >
  > Roll-back of Strata Logging Service integration migration is not supported.

> **PREREQUISITE:**
>
> Configuration of data ingestion from multiple accounts requires Super User permissions in both Cortex XSIAM tenant and Prisma Access accounts.

The logs ingested by Prisma Access are the same as the logs ingested by Next-Generation Firewall. For more information, refer to Ingest data from Next-Generation Firewall.

To ingest detection data from Prisma Access:

1. Select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select Prisma Access, and click Connect.

   > **NOTE:**
   >
   > Cortex XSIAM does not validate your Prisma Access account credentials. You must ensure the account has been deployed in order for data to stream.

3. In the Connect Prisma Access dialog box, you can choose to connect Prisma Access to this account or other accounts.
   - To connect Prisma Access to this account, click Connect.
   - To connect Prisma Access to other accounts, click Connect Prisma Access from other accounts and select the account from the accounts listed. Click Connect.

   Connection can take up to several minutes.

   On the Data Sources page, expand Prisma Access to track the status of your instance.
4. Validate that your data is streaming.

   To ensure the data is streaming into your tenant, using XQL, query Next-Generation Firewall raw datasets `panw_ngfw_<*>_raw` using the field: `is_prisma_mobile`.
5. (Optional) Manage your Instance.

   After you create the Prisma Access instance, on the Data Sources page, expand the Prisma Access integration to track the connection, or, if you want, to Delete the instance.

1.5.4 | Ingest logs from Prisma Access Browser

Abstract

Ingest Prisma Access Browser logs into Cortex XSIAM.

Cortex Prisma Access Browser is a browser designed specifically for enterprise use, and is fortified with security features to protect users and organizations. You can configure Cortex XSIAM to ingest Prisma Access Browser logs into a dataset called `panw_prisma_access_browser_raw`, that can be queried using XQL. This integration gives you visibility into issues that are generated by the browser. The ingested data can also be used for performing threat hunting queries and correlations within the Cortex platform.

Only one instance of this collector can be created per Cortex XSIAM tenant.

1. In Cortex XSIAM, select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select Prisma Access Browser, and click Connect.
3. In the Connect Prisma Access Browser dialog box, select the checkbox for Connect Prisma Access Browser to this account.
4. Click Connect.

   Connection can take up to several minutes.

   On the Data Sources page, expand Prisma Access Browser to track the status of your instance.
5. Validate that data is streaming to your tenant by using XQL to query the dataset `panw_prisma_access_browser_raw`.

After you have created a Prisma Access Browser instance, you can use the Data Sources page to view information about the integration, or delete the instance.

1.5.5 | Ingest detection data from Strata Logging Service

Abstract

Learn how to ingest detection data from Strata Logging Service.

To streamline the connection and management of all Palo Alto Networks generated logs across products in Cortex XSIAM with a Strata Logging Service, Cortex XSIAM can ingest detection data from Strata Logging Service in a more flexible manner using the Strata Logging Service data collector.

You can configure the Strata Logging Service data collector to take logs from other Palo Alto Networks products already logging to 1 or more existing Strata Logging Service.

Cortex XSIAM supports streaming data directly from Prisma Access accounts and New-Generation Firewalls (NGFW) and Panorama devices to your Cortex XSIAM tenants using the Cortex Native Data Lake. Existing integrations should be migrated to the Cortex Native Data Lake. Make sure you select all your devices to connect directly to Cortex XSIAM. Integrations not migrated manually will be migrated automatically 2 weeks before the end of the contract with Strata Logging Service.

For stitched raw data, use the XQL query `xdr_data` dataset or any preset designated for stitched data, such as `network_story`. For query examples, refer to the in-app XQL Library. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC only) when relevant from Strata Logging Service detection data. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

> **NOTE:**
>
> IOC and BIOC issues are applicable on stitched data only and are not available on raw data.

To ingest detection data from Strata Logging Service.

1. Activate the Strata Logging Service.

   You can configure Cortex XSIAM to take Palo Alto generated firewall logs from other Palo Alto Networks products already logging to an existing Strata Logging Service.
2. Select Settings → Data Sources.
3. In the Strata Logging Service configuration, click the more options icon, and select Add New Instance.
4. Select Data Lake Instance.

   Select one or more existing Strata Logging Service instances that you want to connect to this Strata Logging Service instance.
5. Save your Strata Logging Service configuration.

   Once events start to come in, a green check mark appears underneath the Strata Logging Service configuration.
6. (Optional) Manage your Strata Logging Service Collector.

   After you create the Strata Logging Service Collector, you can make additional changes, as needed.

   - Delete the Strata Logging Service Collector.
7. After Cortex XSIAM begins receiving data from a Strata Logging Service, you can use XQL Search to search for specific data, using the `xdr_data` dataset.

1.5.6 | Ingest alerts and assets from IoT Security

Abstract

Ingest alerts and device data from IoT Security.

The Palo Alto Networks IoT Security solution discovers unmanaged devices, detects behavioral anomalies, recommends policy based on risk, and automates enforcement without the need for additional sensors or infrastructure. The Cortex XSIAM IoT Security integration enables you to ingest alerts and device information from your IoT Security instance.

To receive data, configure the Data Sources settings in Cortex XSIAM for the IoT Security data collector in Settings → Data Sources.

As soon as data collection begins, Cortex XSIAM displays the IoT Security alerts in the Cortex XSIAM Issues table and groups them into cases. The IoT Security issues are updated every 15 minutes. IoT security alerts which were resolved before the integration aren’t added to the Cortex XSIAM table. Cortex XSIAM adds device activities detected by IoT Security into the Cortex XSIAM Assets table. Device activities are updated every five minutes.

Cortex XSIAM automatically creates a new dataset for device activities (`panw_iot_security_devices_raw`) and a new dataset for issues (`panw_iot_security_alerts_raw`), which you can use to initiate XQL Search queries and create Correlation Rules.

Before you configure the IoT Security Collector, generate an access key and a key ID for the integration.

1. Log in to the PAN IoT Security portal and click your user name.
2. Select Preferences.
3. In the User Role & Access section, Create an API Access Key.
4. Download and save the access key and key ID in a secure location.

For more information about the PAN IoT Secuity API, see Get Started with the IoT Security API.

Configure the IoT Security alerts and assets collection in Cortex XSIAM.

1. Select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select IoT Security Collector, and click Connect.
3. Specify the following parameters.
   - Customer ID: Tenant domain part of the FQDN used for your IoT Security account. For example, in `yourcorp.iot.paloaltonetworks.com`, the customer ID is `yourcorp`. The customer ID is unique and case sensitive. After you save the integration instance, you can't edit the Customer ID.
   - Access Key and Key ID previously generated for the integration.
   - Integration Scope: Select at least one of the two values, Alerts and Devices depending on which information you want to ingest.
4. Click Test to validate access, and then click Enable.

   When events start to come in, a green check mark appears underneath the IoT Security Collector configuration with the data and time that the data was last synced.
5. (Optional) Manage your IOT Security Collector.

   After you enable the IOT Security Collector, you can make additional changes as needed. To modify a configuration, select any of the following options.
   - Edit the IOT Security Collector settings.
   - Disable the IOT Security Collector.
   - Delete the IOT Security Collector.
6. After Cortex XSIAM begins receiving data from IOT Security, you can use the XQL Search to search for logs in the new datasets, `panw_iot_security_devices_raw` for device activities, and `panw_iot_security_alerts_raw` for issues.

1.5.7 | Collecting URL and File log types

Abstract

Learn about the implications of turning off or on collection of URL and File logs.

For Palo Alto Networks integrations, you can choose whether to collect URL and File type logs. These logs enhance your cyber analytics, correlation rules and visibility for investigation. However, if you want to reduce ingestion charges, you can globally turn off collection of URL and File log types for all Palo Alto Networks Integrations.

When collection is turned off, some detectors won’t detect cyber attacks or provide full context, and correlation rules won’t be able to detect cyber events. For a full list of affected detectors, see Detectors connected to URL and File log types.

You can also calculate the amount of ingestion that URL and File log types are consuming by looking at the NGFW dashboard. This dashboard provides an overview of the PAN-NGFW ingestion status of all log types (including URL and File log types) and their daily consumption quota. For more information, see ???.

You can turn on or off URL and File log types collection on the Data Sources page.

1.5.7.1 | Detectors connected to URL and File log types

Abstract

A list of detectors connected to URL and File log types.

If you turn off URL and File log types collection, some detectors are unable to detect cyber attacks or provide full context, and correlation rules are unable to detect cyber events.

The following detectors are affected by URL logs:

Read more...

- A non-browser process accessed a website UI
- Reverse SSH tunnel to external domain/IP
- Uncommon network tunnel creation
- Suspicious domain fronting behavior
- Possible watering hole SMB credential theft
- Rare connection to external IP address or host by an application using RMI-IIOP or LDAP protocol
- Uncommon JA3 SSL fingerprint communication to an instant messaging server
- PowerShell Initiates a Network Connection to GitHub
- Non-browser failed access to a pastebin-like site
- Non-browser access to a pastebin-like site
- C2 from contextual causality signal
- Massive upload to a rare storage or mail domain
- DNS Tunneling

The following detectors are affected by File logs:

Read more...

- Rare AppID usage to a rare destination
- Abnormal network communication through TOR using an uncommon port
- Recurring access to rare IP
- Possible network connection to a TOR relay server
- A user accessed an uncommon AppID
- Large Upload (Generic)
- Large Upload (FTP)
- Large Upload (SMTP)
- Possible network connection to a TOR relay server
- A user accessed a resource for the first time via SSO - silent
- Access to a domain that is categorized as malicious - silent
- Recurring access to rare domain categorized as malicious - silent
- Cloud Large Upload (Generic) - disabled

1.6 | External data ingestion

Abstract

Cortex XSIAM supports data ingestion from external sources for a variety of service types and vendors.

Cortex XSIAM supports data ingestion from external sources for a variety of service types and vendors.

To ensure complete and uninterrupted data ingestion, Cortex XSIAM collects granular data ingestion metrics and generates ingestion issues if disruption is identified in data collection. For more information, see Overview of data ingestion metrics.

1.6.1 | External applications

Abstract

Learn more about integrating Slack and a Syslog Receiver to Cortex XSIAM.

You can integrate the following external applications to manage notifications:

- Slack: To send outbound notifications to Slack. For more information, see Integrate Slack for outbound notifications.
- Syslog server: To send Cortex XSIAM notifications to your Syslog server. For more information, see Integrate a syslog receiver.

1.6.2 | Ingest network connection logs

Abstract

Cortex XSIAM can ingest network connection logs from different third-party sources.

You can ingest network connection logs from different third-party sources.

1.6.2.1 | Ingest network flow logs from Amazon S3

Abstract

Take advantage of Cortex XSIAM investigation capabilities and set up network flow log ingestion for your Amazon S3 logs using an AWS CloudFormation Script.

You can forward network flow logs to Cortex XSIAM from Amazon Simple Storage Service (Amazon S3).

To receive network flow logs from Amazon S3, you must first configure data collection from Amazon S3. You can then configure the Data Sources settings in Cortex XSIAM for Amazon S3. After you set up collection integration, Cortex XSIAM begins receiving new logs and data from the source.

You can either configure Amazon S3 with SQS notification manually on your own or use the AWS CloudFormation Script that we have created for you to make the process easier. The instructions below explain how to configure Cortex XSIAM to receive network flow logs from Amazon S3 using SQS. To perform these steps manually, see Configure Data Collection from Amazon S3 Manually.

> **NOTE:**
>
> For more information on configuring data collection from Amazon S3, see the Amazon S3 Documentation.

When Cortex XSIAM begins receiving logs, the app automatically creates an Amazon S3 Cortex Query Language (XQL) dataset (`aws_s3_raw`). This enables you to search the logs with XQL Search using the dataset. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to ingest network flow logs as Cortex XSIAM network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC) when relevant from Amazon S3 logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides the following:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

Be sure you do the following tasks before you begin configuring data collection from Amazon S3 using the AWS CloudFormation Script.

- Ensure that you have the proper permissions to run AWS CloudFormation with the script provided in Cortex XSIAM. You need at a minimum the following permissions in AWS for an Amazon S3 bucket and Amazon Simple Queue Service (SQS):
    - Amazon S3 bucket: `GetObject`
    - SQS: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`.
- Ensure that you can access your Amazon Virtual Private Cloud (VPC) and have the necessary permissions to create flow logs.
- Determine how you want to provide access to Cortex XSIAM to your logs and perform API operations. You have the following options:
    - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user. This is the default option as explained in Configure the Amazon S3 Collection in Cortex XSIAM by selecting Access Key.
    - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. For more information, see Creating a role to delegate permissions to an AWS service. This is the Assumed Role option as described in the Configure the Amazon S3 collection in Cortex XSIAM. For more information on creating an assumed role for Cortex XSIAM, see Create an assumed role.

  To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

Configure Cortex XSIAM to receive network flow logs from Amazon S3 using the CloudFormation Script.

1. Download the CloudFormation Script in Cortex XSIAM.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Amazon S3, and click Connect.

    c. To provide access to Cortex XSIAM to your logs and to perform API operations using a designated AWS IAM user, leave the Access Key option selected. Otherwise, select Assumed Role, and ensure that you Create an Assumed Role for before continuing with these instructions.

    d. For the Log Type, select Flow Logs to configure your log collection to receive network flow logs from Amazon S3, and the following text is displayed under the field Download CloudFormation Script. See instructions here.

    e. Click the Download CloudFormation Script. link to download the script to your computer.

2. Create a new Stack in the CloudFormation Console with the script you downloaded from Cortex XSIAM.

    For more information on creating a Stack, see Creating a stack on the AWS CloudFormation console.

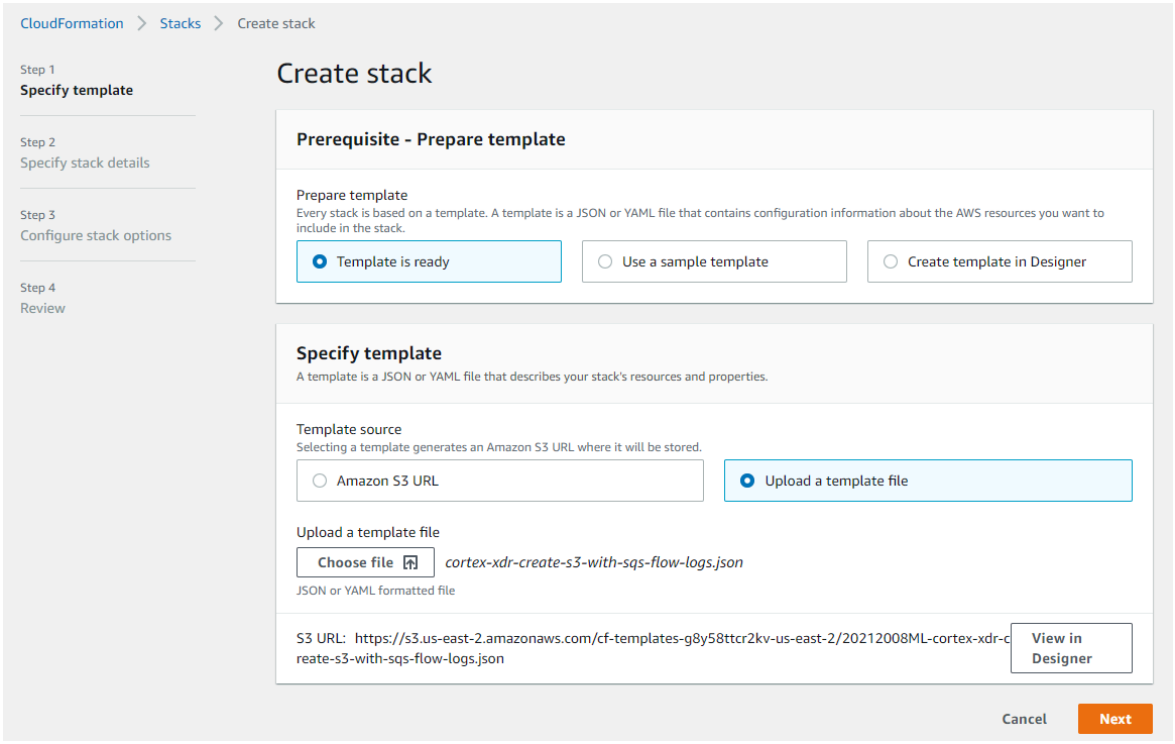
    a. Log in to the CloudFormation Console.

    b. From the CloudFormation → Stacks page, ensure that you have selected the correct region for your configuration.

    c. Select Create Stack → With new resources (standard).

    d. Specify the template that you want AWS CloudFormation to use to create your stack. This template is the script that you downloaded from Cortex XSIAM , which will create an Amazon S3 bucket, Amazon Simple Queue Service (SQS) queue, and Queue Policy. Configure the following settings in the Specify template page.

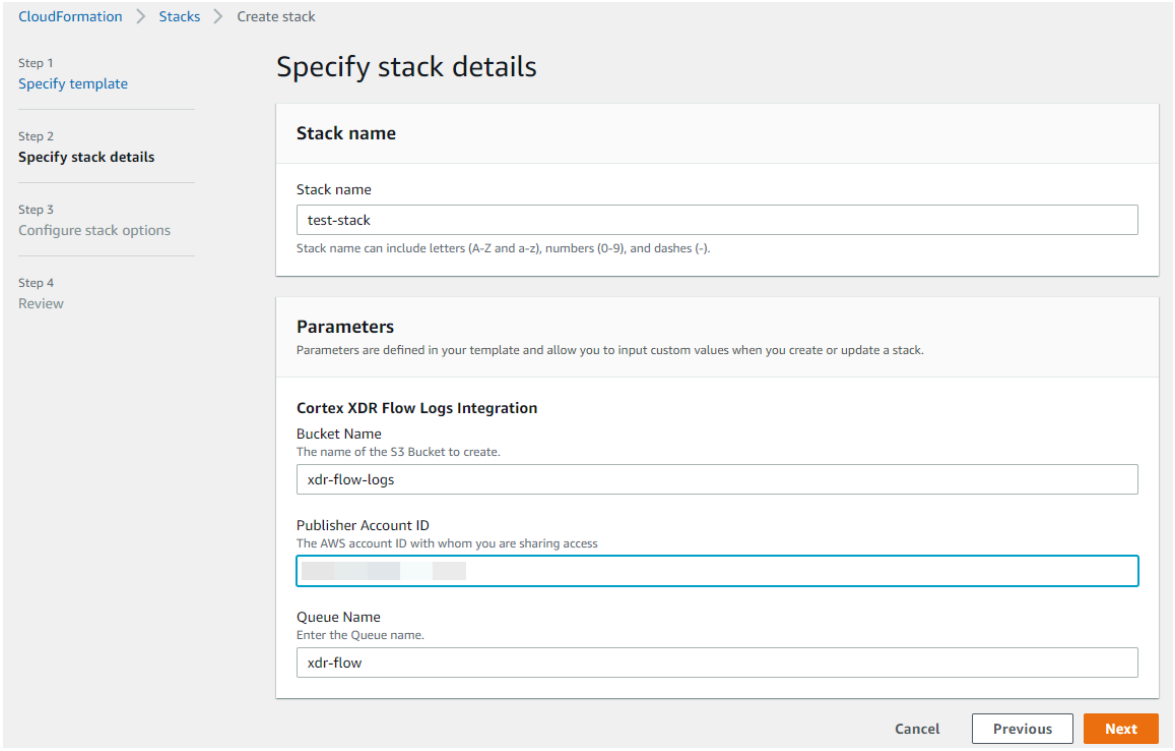
- Prerequisite - Prepare template → Prepare template: Select Template is ready.
- Specify Template
  - Template source: Select Upload a template file.
  - Upload a template file: Choose file, and select the `cortex-xdr-create-s3-with-sqs-flow-logs.json` file that you downloaded from Cortex XDR.



e. Click Next.

f. In the Specify stack details page, configure the following stack details.

- Stack name: Specify a descriptive name for your stack.
- Parameters → Cortex XDR Flow Logs Integration
  - Bucket Name: Specify the name of the S3 bucket to create, where you can leave the default populated name as xdr-flow-logs or create a new one. The name must be unique.
  - Publisher Account ID: Specify the AWS IAM user account ID with whom you are sharing access.
  - Queue Name: Specify the name for your Amazon SQS queue to create, where you can leave the default populated name as xdr-flow or create a new one. The name must be unique.



g. Click Next.

h. In the Configure stack options page, there is nothing to configure, so click Next.

i. In the Review page, look over the stack configurations settings that you have configured and if they are correct, click Create stack. If you need to make a change, click Edit beside the particular step that you want to update.

The stack is created and is opened with the Events tab displayed. It can take a few minutes for the new Amazon S3 bucket, SQS queue, and Queue Policy to be created. Click Refresh to get updates. Once everything is created, leave the stack opened in the current browser, because you will need to access information in the stack for other steps detailed below.

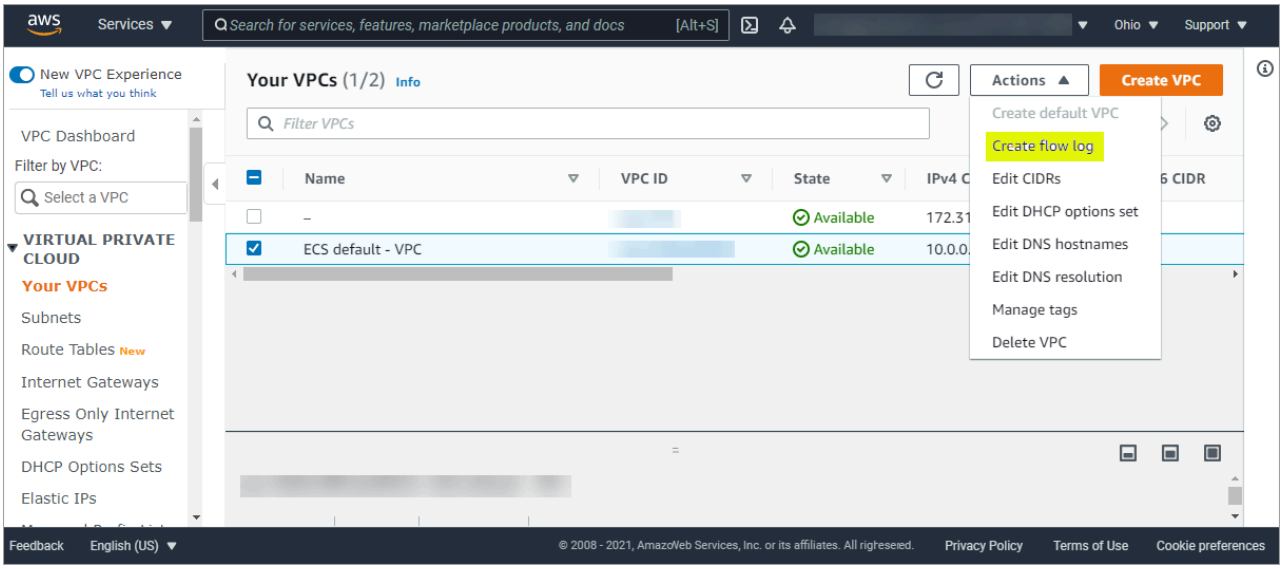
> **NOTE:**
>
> For the Amazon S3 bucket created using CloudFormation, it is the customer's responsibility to define a retention policy by creating a Lifecycle rule in the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.

3. Configure your Amazon Virtual Private Cloud (VPC) with flow logs:

1. Open the Amazon VPC Console, and in the Resources by Region listed, select VPCs to view the VPCs configured for the current region selected. To select another VPC from another region, select See all regions, and select one of them.

> **NOTE:**
>
> To create a new VPC, click Launch VPC Wizard. For more information, see AWS VPC Flow Logs.

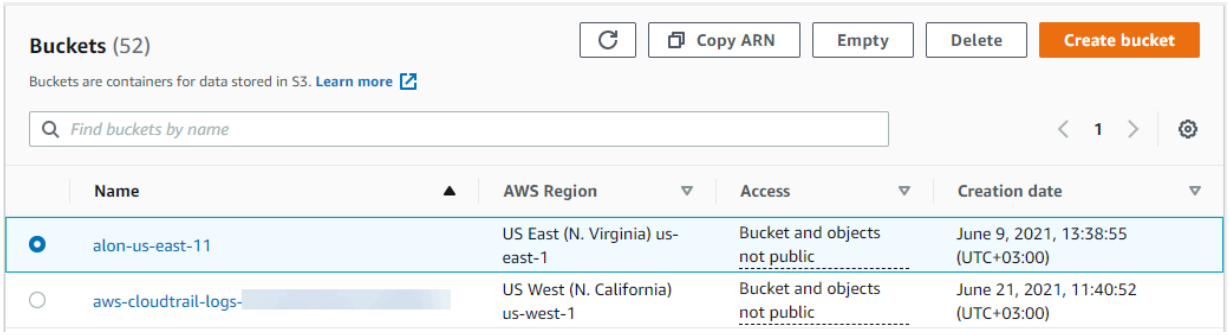
2. From the list of Your VPCs, select the checkbox beside the VPC that you want to configure to create flow logs, and then select Actions → Create flow log.



3. Configure the following Flow log settings:

- Name - optional: (Optional) Specify a descriptive name for your VPC flow log.
- Filter: Select All types of traffic to capture.
- Maximum aggregation interval: If you anticipate a heavy flow of traffic, select 1 minute. Otherwise, leave the default setting as 10 minutes.
- Destination: Select Send to an Amazon S3 bucket as the destination to publish the flow log data.
- S3 bucket ARN:Specify the Amazon Resource Name (ARN) for your Amazon S3 bucket.

  You can retrieve your bucket's ARN by opening another instance of the AWS Management Console in a browser window and opening the Amazon S3 console. In the Buckets section, select the bucket that you created for collecting the Amazon S3 flow logs when you created your stack, click Copy ARN, and paste the ARN in this field.



- Log record format: Select Custom Format, and in the Log Format field, specify the following fields to include in the flow log record, which you can select from the list displayed:

- account-id
- action
- az-id
- bytes
- dstaddr
- dstport
- end
- flow-direction
- instance-id
- interface-id
- packets
- log-status
- pkt-srcaddr
- pkt-dstaddr
- protocol
- region
- srcaddr
- srcport
- start
- sublocation-id
- sublocation-type
- subnet-id
- tcp-flags
- type
- vpc-id
- version

4. Click Create flow log.

   Once the flow log is created, a message indicating that the flow log was successfully created is displayed at the top of the Your VPCs page.

   In addition, if you open your Amazon S3 bucket configurations, by selecting the bucket from the Amazon S3 console, the Objects tab contains a folder called `AWSLogs/` to collect the flow logs.

4. Configure access keys for the AWS IAM user that Cortex XSIAM uses for API operations.

> **NOTE:**
>
> - It is the responsibility of the customer's organization to ensure that the user who performs this task of creating the access key is designated with the relevant permissions. Otherwise, this can cause the process to fail with errors.
> - Skip this step if you are using an Assumed Role for Cortex XSIAM.

1. Open the AWS IAM Console, and in the navigation pane, select Access management → Users.
2. Select the User name of the AWS IAM user.
3. Select the Security credentials tab, scroll down to the Access keys section, and click Create access key.
4. Click the copy icon next to the Access key ID and Secret access key keys, where you must click Show secret access key to see the secret key and record them somewhere safe before closing the window. You will need to provide these keys when you edit the Access policy of the SQS queue and when setting the AWS Client ID and AWS Client Secret in Cortex XSIAM. If you forget to record the keys and close the window, you will need to generate new keys and repeat this process.

> **NOTE:**

> For more information, see Managing access keys for IAM users.

5. When you create an Assumed Role in Cortex XSIAM, ensure that you edit the policy that defines the permissions for the role with the S3 Bucket ARN and SQS ARN, which is taken from the Stack you created.

   > **NOTE:**
   >
   > Skip this step if you are using an Access Key to provide access to Cortex XSIAM.

6. Configure the Amazon S3 collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. In the Amazon S3 configuration, click Add Instance to begin a new configuration.

   c. Set these parameters, where the parameters change depending on whether you configured an Access Key or Assumed Role.

   - SQS URL: Specify the SQS URL, which is taken from the stack you created. In the browser you left open after creating the stack, open the Outputs tab, copy the Value of the QueueURL and paste it in this field.
   - Name: Specify a descriptive name for your log collection configuration.
   - When setting an Access Key, set these parameters.
     - AWS Client ID: Specify the Access key ID, which you received when you created access keys for the AWS IAM user in AWS.
     - AWS Client Secret: Specify the Secret access key you received when you created access keys for the AWS IAM user in AWS.
   - When setting an Assumed Role, set these parameters.
     - Role ARN: Specify the Role ARN for the Assumed Role you created for in AWS.
     - External Id:Specify the External Id for the Assumed Role you created for in AWS.
   - Log Type: Select Flow Logs to configure your log collection to receive network flow logs from Amazon S3. When configuring network flow log collection, the following additional field is displayed for Enhanced Cloud Protection.

     You can Normalize and enrich flow logs by selecting the checkbox. If selected, Cortex XSIAM ingests the network flow logs as XDR network connection stories, which you can query using XQL Search from the `xdr_data` dataset using the preset called `network_story`.

   d. Click Test to validate access, and then click Enable.

   When events start to come in, a green check mark appears underneath the Amazon S3 configuration with the number of logs received.

1.6.2.1.1 | Create an assumed role

Abstract

Learn about creating an AWS Assumed Role for Cortex XSIAM.

If you do not designate a separate AWS IAM user to provide access to Cortex XSIAM to your logs and to perform API operations, you can create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your logs. For more information, see Creating a role to delegate permissions to an AWS service.

When setting up any type of Amazon S3 Collector in Cortex XSIAM, these instructions explain setting up an Assumed Role.

1. Log in to the AWS Management Console to create a role for Cortex XSIAM.

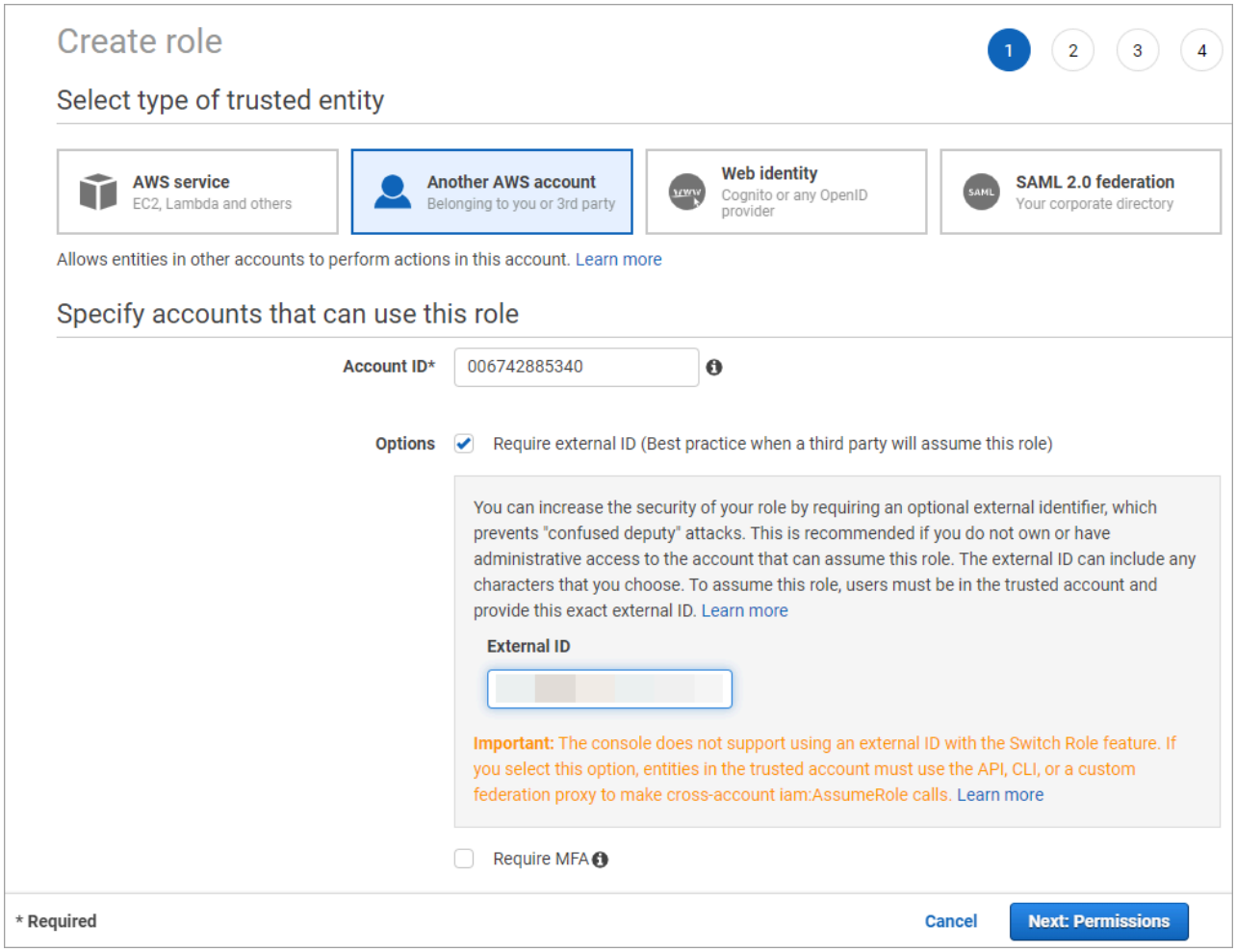
   Refer to the AWS instructions for guidance.

   a. Create the role in the same region as your AWS account, and use the following values and options when creating the role.

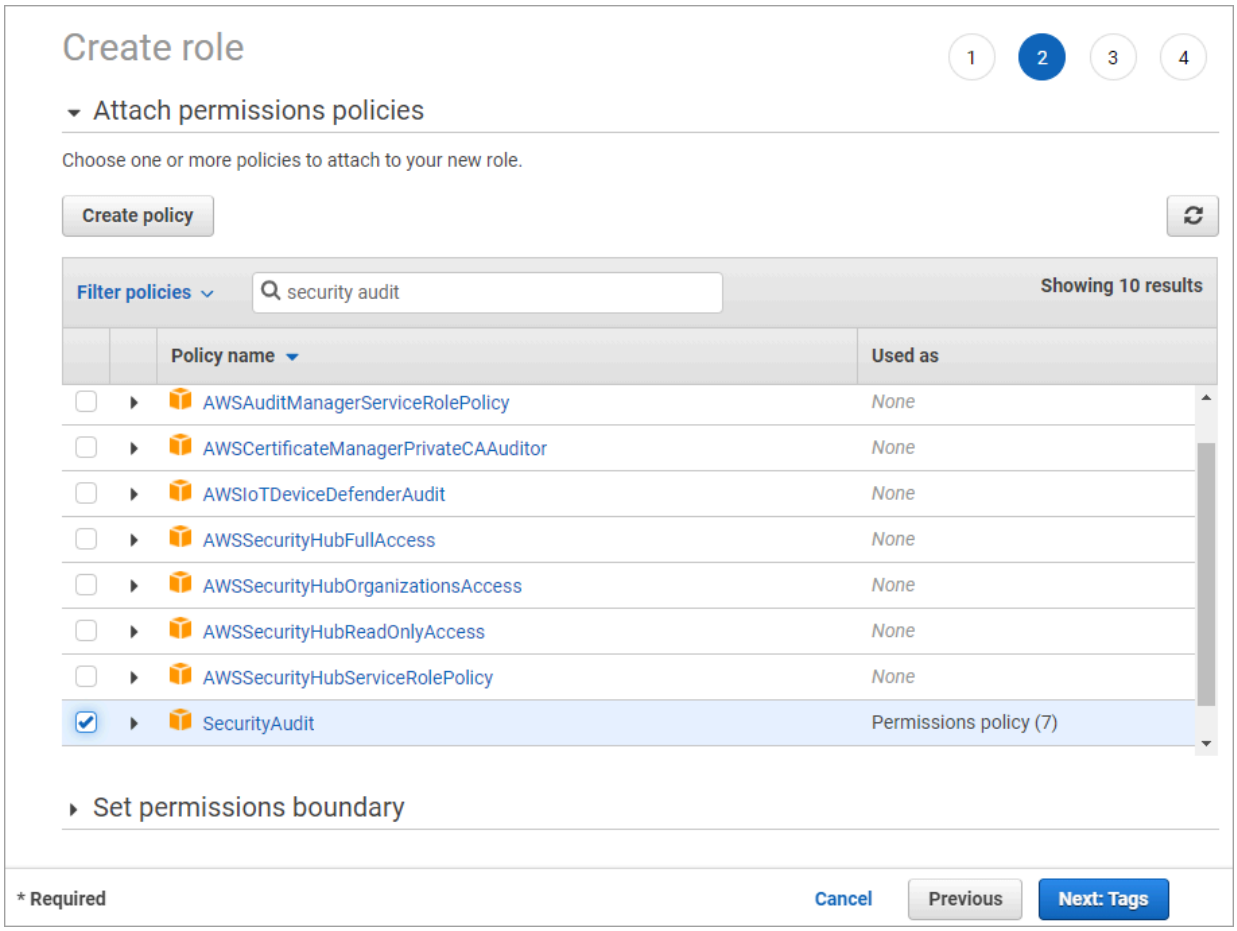
   - Type of Trusted → Another AWS Account, and specify the Account ID as `006742885340`. When using a Cortex XSIAM FedRAMP environment, specify the Account ID as `685269782068`.
   - Select Options for the Require external ID, which is a unique alphanumeric string, and generate a secure UUIDv4 using an Online UUID Generator. Copy the External ID as you will use this when configuring the Amazon S3 Collector in Cortex XSIAM .

     > **NOTE:**
     >
     > In AWS this is an optional field to configure, but this must be configured to set up the Amazon S3 Collector in Cortex XSIAM .

   - Do not enable MFA. Verify that Require MFA is not selected.



b. Click Next and add the AWS Managed Policy for Security Audit.



Then, add a role name and create the role. In this workflow, later, you will create the granular policies and edit the role to attach the additional policies.

2. Create the policy that defines the permissions for the Cortex XSIAM role.

   a. Select IAM on the AWS Management Console.

   b. In the navigation pane on the left, select Access Management → Policies → Create Policy.

   c. Select the JSON tab.

   Copy the following JSON policy and paste it within the editor window.

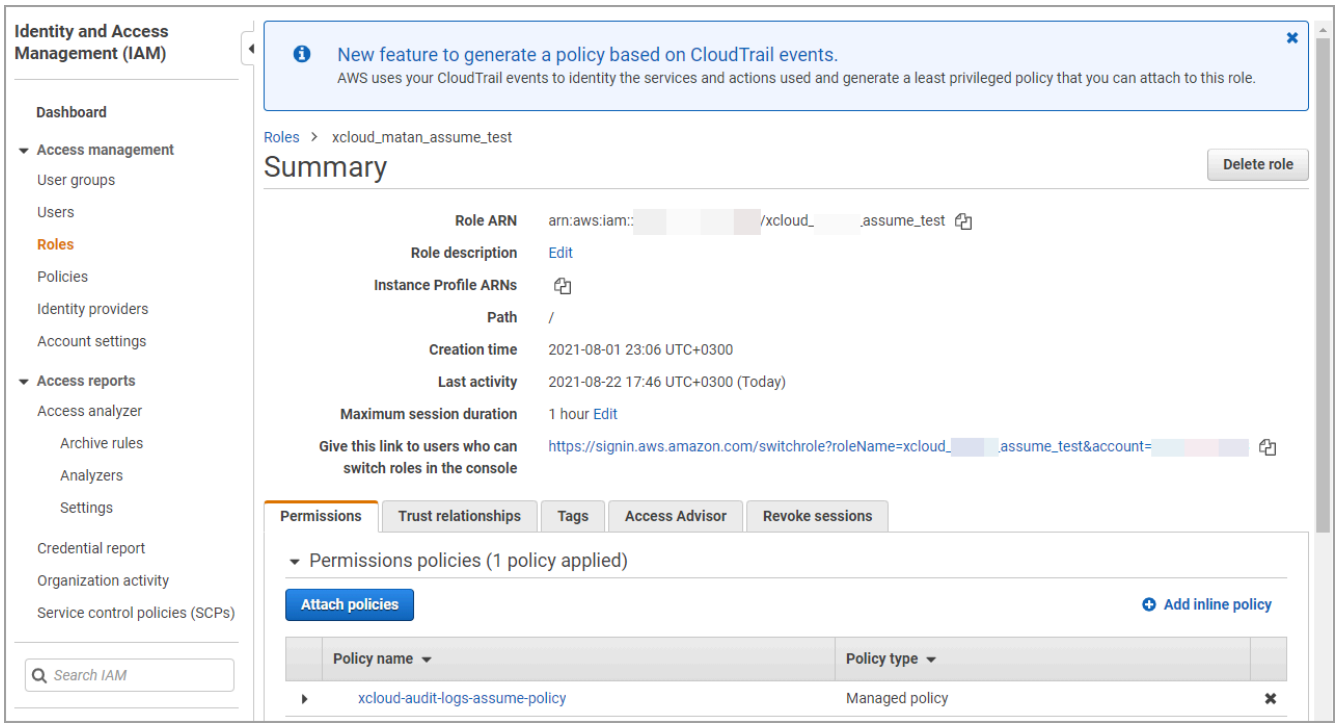
   > **NOTE:**
   >
   > The **<s3-arn>** and **<sqs-arn>** placeholders. These will be filled out later depending on which Amazon S3 logs you are configuring, including network flow logs, audit logs, or generic logs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "<s3-arn>/*"
        },
        {
            "Effect": "Allow",
             "Action": [
                "sqs:ReceiveMessage",
                "sqs:DeleteMessage",
                "sqs:ChangeMessageVisibility"
            ],
            "Resource": "<sqs-arn>"
        }
    ]
}
```

   d. Review and create the policy.

3. Edit the role you created in Step 1 and attach the policy to the role.

4. Copy the Role ARN.



5. Continue with the task for the applicable Amazon S3 logs you want to configure.

The following type of logs are available.

- Ingest network flow logs from Amazon S3.
- Ingest network Route 53 logs from Amazon S3
- Ingest audit logs from AWS Cloud Trail.
- Ingest generic logs from Amazon S3.

1.6.2.1.2 | Configure data collection from Amazon S3 manually

Abstract

Set up network flow log ingestion for your Amazon S3 logs manually (without a script).

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Runtime Security or Data Collection add-on.

There are various reasons why you may need to configure data collection from Amazon S3 manually, as opposed to using the CloudFormation Script provided in Cortex XSIAM. For example, if your organization does not use CloudFormation scripts, you will need to follow the instructions below, which explain at a high-level how to perform these steps manually with a link to the relevant topic in the Amazon S3 documentation with the detailed steps to follow.

As soon as Cortex XSIAM begins receiving logs, the app automatically creates an Amazon S3 Cortex Query Language (XQL) dataset (`aws_s3_raw`). This enables you to search the logs with XQL Search using the dataset. For example queries, refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to ingest network flow logs as Cortex XSIAM network connection stories, which you can query with XQL Search using the `xdr_dataset` dataset with the preset called `network_story`. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlations, IOC, and BIOC) when relevant from Amazon S3 logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

Be sure you do the following tasks before you begin configuring data collection manually from Amazon CloudWatch to Amazon S3.

> **NOTE:**
>
> If you already have an Amazon S3 bucket configured with VPC flow logs that you want to use for this configuration, you do not need to perform the prerequisite steps detailed in the first two bullets.

- Ensure that you have at a minimum the following permissions in AWS for an Amazon S3 bucket and Amazon Simple Queue Service (SQS).
  - Amazon S3 bucket: `GetObject`
  - SQS: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`.
- Create a dedicated Amazon S3 bucket for collecting network flow logs with the default settings. For more information, see Creating a bucket using the Amazon S3 Console.

  > **NOTE:**
  >
  > It is your responsibility to define a retention policy for your Amazon S3 bucket by creating a Lifecycle rule in the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.
- Ensure that you can access your Amazon Virtual Private Cloud (VPC) and have the necessary permissions to create flow logs.
- Determine how you want to provide access to Cortex XSIAM to your logs and perform API operations. You have the following options.
  - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user. This is the default option as explained in Configure the Amazon S3 collection by selecting Access Key.
  - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. For more information, see Creating a role to delegate permissions to an AWS service. This is the Assumed Role option as described in the Configure the Amazon S3 collection. For more information on creating an assumed role for Cortex XSIAM , see Create an assumed role.

  To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

Configure Cortex XSIAM to receive network flow logs from Amazon S3 manually.

1. Log in to the AWS Management Console.
2. From the menu bar, ensure that you have selected the correct region for your configuration.
3. Configure your Amazon Virtual Private Cloud (VPC) with flow logs. For more information, see AWS VPC Flow Logs.

   > **NOTE:**
   >
   > If you already have an Amazon S3 bucket configured with VPC flow logs, skip this step and go to Configure an Amazon Simple Queue Service (SQS).

4. Configure an Amazon Simple Queue Service (SQS). For more information, see Configuring Amazon SQS queues (console).

   > **NOTE:**
   >
   > Ensure that you create your Amazon S3 bucket and Amazon SQS queue in the same region.

5. Configure an event notification to your Amazon SQS whenever a file is written to your Amazon S3 bucket. For more information, see Amazon S3 Event Notifications.
6. Configure access keys for the AWS IAM user that Cortex XSIAM uses for API operations. For more information, see Managing access keys for IAM users.

   > **NOTE:**
   >
   > - It is the responsibility of the customer's organization to ensure that the user who performs this task of creating the access key is designated with the relevant permissions. Otherwise, this can cause the process to fail with errors.
   > - Skip this step if you are using an Assumed Role for Cortex XSIAM.

7. Update the Access Policy of your SQS queue and grant the required permissions mentioned above to the relevant IAM user. For more information, see Granting permissions to publish event notification messages to a destination.

   > **NOTE:**
   >
   > Skip this step if you are using an Assumed Role for Cortex XSIAM.

8. Configure the Amazon S3 collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Amazon S3, and click Connect.

   c. Set these parameters, where the parameters change depending on whether you configured an Access Key or Assumed Role.

   - To provide access to Cortex XSIAM to your logs and perform API operations using a designated AWS IAM user, leave the Access Key option selected. Otherwise, select Assumed Role, and ensure that you Create an Assumed Role for Cortex XSIAM before continuing with these instructions. In addition, when you create an Assumed Role for Cortex XSIAM, ensure that you edit the policy that defines the permissions for the role with the Amazon S3 Bucket ARN and SQS ARN.
   - SQS URL: Specify the SQS URL, which is the ARN of the Amazon SQS that you configured in the AWS Management Console. For more information on how to retrieve your Amazon SQS ARN, see the Specify SQS queue field when you configure an event notification to your Amazon SQS whenever a file is written to your Amazon S3 bucket.
   - Name: Specify a descriptive name for your log collection configuration.
   - When setting an Access Key, set these parameters.
     - AWS Client ID: Specify the Access key ID, which you received when you created access keys for the AWS IAM user in AWS.
     - AWS Client Secret: Specify the Secret access key you received when you created access keys for the AWS IAM user in AWS.
   - When setting an Assumed Role, set these parameters.
     - Role ARN: Specify the Role ARN for the Assumed Role for Cortex XSIAM in AWS.
     - External Id: Specify the External Id for the Assumed Role for Cortex XSIAM in AWS.
   - Log Type: Select Flow Logs to configure your log collection to receive network flow logs from Amazon S3. When configuring network flow log collection, the following additional field is displayed for Enhanced Cloud Protection.

     You can Normalize and enrich flow logs by selecting the checkbox. When selected, Cortex XSIAM ingests the network flow logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset using the preset called `network_story`.

   d. Click Test to validate access, and then click Enable.

   Once events start to come in, a green check mark appears underneath the Amazon S3 configuration with the number of logs received.

1.6.2.2 | Ingest network Route 53 logs from Amazon S3

Abstract

Take advantage of Cortex XSIAM investigation capabilities and set up network Route 53 ingestion for your Amazon S3 logs using an AWS CloudFormation Script.

You can forward network AWS Route 53 DNS logs to Cortex XSIAM from Amazon Simple Storage Service (Amazon S3).

To receive network Route 53 DNS logs from Amazon S3, you must first configure data collection from Amazon S3. You can then configure the Collection Integrations settings in Cortex XSIAM for Amazon S3. After you set up collection integration, Cortex XSIAM begins receiving new logs and data from the source.

You can configure Amazon S3 with SQS notification using the AWS CloudFormation Script that we have created for you to make the process easier. The instructions below explain how to configure Cortex XSIAM to receive network Route 53 DNS logs from Amazon S3 using SQS.

> **NOTE:**
>
> For more information on configuring data collection from Amazon S3 for Route 53 DNS logs, see the AWS Documentation.

When Cortex XSIAM begins receiving logs, the app automatically creates an Amazon Route 53 Cortex Query Language (XQL) dataset (`amazon_route53_raw`). This enables you to search the logs with XQL Search using the dataset. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to ingest network Route 53 DNS logs as Cortex XSIAM network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC) when relevant from Amazon Route 53 DNS logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

Be sure you do the following tasks before you begin configuring data collection from Amazon S3 using the AWS CloudFormation Script.

- Ensure that you have the proper permissions to run AWS CloudFormation with the script provided in Cortex XSIAM. You need at a minimum the following permissions in AWS for an Amazon S3 bucket and Amazon Simple Queue Service (SQS):
  - Amazon S3 bucket: `GetObject`
  - SQS: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`.
- Ensure that you can access your Amazon Virtual Private Cloud (VPC) and have the necessary permissions to create Route 53 Resolver Query logs.
- Determine how you want to provide access to Cortex XSIAM to your logs and perform API operations. You have the following options.
  - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user. This is the default option when you configure the Amazon S3 collection by selecting Access Key.
  - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. For more information, see Creating a role to delegate permissions to an AWS service. This is the Assumed Role option when you configure the Amazon S3 collection in Cortex XSIAM. For more information on creating an assumed role for Cortex XSIAM, see Create an assumed role.

  To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

Configure Cortex XSIAM to receive network Route 53 DNS logs from Amazon S3 using the CloudFormation Script.

1. Download the CloudFormation Script in Cortex XSIAM.
   a. Select Settings → Data Sources.
   b. On the Data Sources page, click Add Data Source, search for and select Amazon S3, and click Connect.
   c. To provide access to Cortex XSIAM to your logs and to perform API operations using a designated AWS IAM user, leave the Access Key option selected. Otherwise, select Assumed Role, and ensure that you Create an Assumed Role for before continuing with these instructions.
   d. For the Log Type, select Route 53 to configure your log collection to receive network Route 53 DNS logs from Amazon S3, and the following text is displayed under the field Download CloudFormation Script. See instructions here.

e. Click the Download CloudFormation Script. link to download the script to your computer.

2. Create a new Stack in the CloudFormation Console with the script you downloaded from Cortex XSIAM.

> **NOTE:**
>
> For more information on creating a Stack, see Creating a stack on the AWS CloudFormation console.

a. Log in to the CloudFormation Console.

b. From the CloudFormation → Stacks page, ensure that you have selected the correct region for your configuration.

c. Select Create Slack → With new resources (standard).

d. Specify the template that you want AWS CloudFormation to use to create your stack. This template is the script that you downloaded from Cortex XSIAM, which will create an Amazon S3 bucket, Amazon Simple Queue Service (SQS) queue, and Queue Policy. Configure the following settings in the Specify template page.

- Prerequisite - Prepare template → Prepare template: Select Template is ready.
- Specify Template
  - Template source: Select Upload a template file.
  - Upload a template file: Choose file, and select the `CloudFormation-Script.json` file that you downloaded.

e. Click Next.

f. In the Specify stack details page, configure the following stack details.

- Stack name: Specify a descriptive name for your stack.
- Parameters → Cortex XDR Flow Logs Integration
  - Bucket Name: Specify the name of the S3 bucket to create, where you can leave the default populated name as xdr-route53-logs or create a new one. The name must be unique.
  - Publisher Account ID: Specify the AWS IAM user account ID with whom you are sharing access.
  - Queue Name: Specify the name for your Amazon SQS queue to create, where you can leave the default populated name as xdr-route53 or create a new one. The name must be unique.

g. Click Next.

h. In the Configure stack options page, there is nothing to configure, so click Next.

i. In the Review page, look over the stack configurations settings that you have configured and if they are correct, click Create stack. If you need to make a change, click Edit beside the particular step that you want to update.

The stack is created and is opened with the Events tab displayed. It can take a few minutes for the new Amazon S3 bucket, SQS queue, and Queue Policy to be created. Click Refresh to get updates. Once everything is created, leave the stack opened in the current browser as you will need to access information in the stack for other steps detailed below.

> **NOTE:**
>
> For the Amazon S3 bucket created using CloudFormation, it is the customer’s responsibility to define a retention policy by creating a Lifecycle rule in the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.

3. Configure Route 53 Query Logging in AWS.

a. Log in to the AWS Management Console.

b. From the menu bar, ensure that you have selected the correct region for your configuration.

c. Search for Route 53 and select Resolver → Query Logging.

d. Configure query logging.

e. Set the following parameters in the different sections on the Configure query logging page.

- Query logging configuration name
    - Name: Specify a name for your Resolver query logging configuration.
- Query logs destination
    - Destination for query logs: Select S3 bucket as the place where you want Resolver to publish query logs.
    - Amazon S3 bucket: Browse S3 to select the Amazon S3 bucket created after running the CloudFormation script, which is by default called xdr-route53-logs or select the one that you created.
- VPCs to log queries for
    - Add VPC: Clicking the Add VPC button opens the Add VPC page, where you can choose the VPCs that you want to log queries for. When you are done, click Add.

f. Click Configure query logging.

4. Configure access keys for the AWS IAM user that Cortex XSIAM uses for API operations.

> **NOTE:**
>
> - It is the responsibility of the customer's organization to ensure that the user who performs this task of creating the access key is designated with the relevant permissions. Otherwise, this can cause the process to fail with errors.
> - Skip this step if you are using an Assumed Role for Cortex XSIAM.

a. Open the AWS IAM Console, and in the navigation pane, select Access management → Users.

b. Select the User name of the AWS IAM user.

c. Select the Security credentials tab, scroll down to the Access keys section, and click Create access key.

d. Click the copy icon next to the Access key ID and Secret access key keys, where you must click Show secret access key to see the secret key and record them somewhere safe before closing the window. You will need to provide these keys when you edit the Access policy of the SQS queue and when setting the AWS Client ID and AWS Client Secret in Cortex XSIAM. If you forget to record the keys and close the window, you will need to generate new keys and repeat this process.

> **NOTE:**
>
> For more information, see Managing access keys for IAM users.

5. When you create an Assumed Role, ensure that you edit the policy that defines the permissions for the role with the S3 Bucket ARN and SQS ARN, which is taken from the stack you created.

> **NOTE:**
>
> Skip this step if you are using an Access Key to provide access to Cortex XSIAM.

6. Configure the Amazon S3 collection in Cortex XSIAM.

a. Select Settings → Data Sources.

b. In the Amazon S3 configuration, click Add Instance to begin a new configuration.

c. Set these parameters, where the parameters change depending on whether you configured an Access Key or Assumed Role.

- SQS URL: Specify the SQS URL, which is taken from the stack you created. In the browser you left open after creating the stack, open the Outputs tab, copy the Value of the QueueURL and paste it in this field.
- Name: Specify a descriptive name for your log collection configuration.
- When setting an Access Key, set these parameters.
    - AWS Client ID: Specify the Access key ID, which you received when you created access keys for the AWS IAM user in AWS.
    - AWS Client Secret: Specify the Secret access key you received when you created access keys for the AWS IAM user in AWS.
- When setting an Assumed Role, set these parameters.
    - Role ARN: Specify the Role ARN for the Assumed Role you created for Cortex XSIAMin AWS.
    - External Id: Specify the External Id for the Assumed Role you created for Cortex XSIAM in AWS.
- Log Type: Select Route 53 to configure your log collection to receive network Route 53 DNS logs from Amazon S3. When configuring network Route 53 log collection, the following additional field is displayed for Enhanced Cloud Protection.

  You can Normalize DNS logs by selecting the checkbox (default configuration). When selected, Cortex XSIAM ingests the network Route 53 DNS logs as XDR network connection stories, which you can query using XQL Search from the `xdr_data` dataset using the preset called `network_story`.

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears underneath the Amazon S3 configuration with the number of logs received.

1.6.2.3 | Ingest logs from Check Point firewalls

Abstract

To take advantage of Cortex XSIAM investigation and detection capabilities while using Check Point firewalls, forward your firewall logs to Cortex XSIAM.

If you use Check Point FW1/VPN1 firewalls, you can still take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your Check Point firewall logs to Cortex XSIAM. Check Point firewall logs can be used as the sole data source, however, you can also use Check Point firewall logs in conjunction with Palo Alto Networks firewall logs and additional data sources.

Cortex XSIAM can stitch data from Check Point firewalls with other logs to make up network stories searchable in the Query Builder and in Cortex Query Language (XQL) queries. Cortex XSIAM can also return raw data from Check Point firewalls in XQL queries.

> **NOTE:**
>
> - Logs with `sessionid = 0` are dropped.
> - Destination Port data is available only in the raw logs.

In terms of alerts, Cortex XSIAM can both surface native Check Point firewall alerts and generate its own issues on network activity. Issues are displayed throughout Cortex XSIAM issue, case, and investigation views.

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog Collector. You then configure your Check Point firewall policy to log all traffic and set up the Log Exporter on your Check Point Log Server to forward logs to the Syslog Collector in a CEF format.

When Cortex XSIAM starts to receive logs, the app can begin stitching network connection logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to generate Analytics issues, and can apply IOC, BIOC, and Correlation Rule matching. You can also use queries to search your network connection logs.

1. Ensure that your Check Point firewalls meet the following requirements.

   Check Point software version: R77.30, R80.10, R80.20, R80.30, or R80.40

2. Increase log storage for Check Point firewall logs.

   As an estimate for initial sizing, note that the average Check Point log size is roughly 700 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Check Point firewalls. For more information, see Manage Your Log Storage within Cortex XSIAM.

3. Activate the Syslog Collector.

4. Configure the Check Point firewall to forward Syslog events in CEF format to the Syslog Collector.

   Configure your firewall policy to log all traffic and set up the Log Exporter to forward logs to the Syslog Collector. For more information on setting up Log Exporter, see the Check Point documentation.

1.6.2.4 | Ingest logs from Cisco ASA firewalls and AnyConnect

Abstract

Extend Cortex XSIAM visibility into logs from Cisco ASA firewalls and Cisco AnyConnect VPN.

If you use Cisco ASA firewalls or Cisco AnyConnect VPN, you can take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your firewall and AnyConnect VPN logs to Cortex XSIAM. This enables Cortex XSIAM to examine your network traffic to detect anomalous behavior. Cortex XSIAM can use Cisco ASA firewall logs and AnyConnect VPN logs as the sole data source, but can also use Cisco ASA firewall logs in conjunction with Palo Alto Networks firewall logs. For additional endpoint context, you can also use Cortex XSIAM to collect and alert on endpoint data.

When Cortex XSIAM starts to receive logs, the app can begin stitching network connection logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to generate Analytics issues, and can apply IOC, BIOC, and Correlation Rules matching. You can also use queries to search your network connection logs using the Cisco Cortex Query Language (XQL) dataset (`cisco_asa_raw`).

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog Collector. You then configure forwarding on your log devices to send logs to the Syslog Collector in a CISCO format.

1. Verify that your Cisco ASA firewall and Cisco AnyConnect VPN logs meet the following requirements.

- Syslog in Cisco-ASA format
- Must include `timestamps`
- Only supports the following messages.
  - For Cisco ASA firewall: 302013, 302014, 302015, 302016
  - For Cisco AnyConnect VPN: 113039, 716001, 722022, 722033, 722034, 722051, 722055, 722053, 113019, 716002, 722023, 722037

2. Activate the Syslog Collector.
3. Increase log storage for Cisco ASA firewall and Cisco AnyConnect VPN logs.

   As an estimate for initial sizing, note that the average Cisco ASA log size is roughly 180 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Cisco ASA firewalls and Cisco AnyConnect VPN logs. For more information, see Manage Your Log Storage within Cortex XSIAM.
4. Configure the Cisco ASA firewall and Cisco AnyConnect VPN, or the log devices forwarding logs from Cisco, to log to the Syslog Collector in a CISCO format.

   Configure your firewall and AnyConnect VPN policies to log all traffic and forward the traffic logs to the Syslog Collector in a CISCO format. By logging all traffic, you enable Cortex XSIAM to detect anomalous behavior from Cisco ASA firewall logs and Cisco AnyConnect VPN logs. For more information on setting up Log Forwarding on Cisco ASA firewalls or Cisco AnyConnect VPN, see the Cisco ASA Series documentation.

1.6.2.5 | Ingest logs from Corelight Zeek

Abstract

Extend Cortex XSIAM visibility into logs from Corelight Zeek.

If you use Corelight Zeek sensors for network monitoring, you can still take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your network connection logs to Cortex XSIAM. This enables Cortex XSIAM to examine your network traffic to detect anomalous behavior. Cortex XSIAM can use Corelight Zeek logs as the sole data source, but can also use logs in conjunction with Palo Alto Networks or third-party firewall logs. For additional endpoint context, you can also use Cortex XSIAM to collect and alert on endpoint data.

As soon as Cortex XSIAM starts to receive logs, the app can begin stitching network connection logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to generate Analytics issues, and can apply IOC, BIOC, and Correlation Rule matching. You can also use queries to search your network connection logs.

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog Collector. You then configure forwarding on your Corelight Zeek sensors (using the default Syslog export option of RFC5424 over TCP) to send logs to the Syslog Collector.

1. Activate the Syslog Collector.

   During activation, you define the Listening Port over which you want the Syslog Collector to receive logs. You must also set TCP as the transport Protocol and Corelight as the Syslog Format.
2. Increase log storage for Corelight Zeek logs.

   For proper sizing calculations, test the log sizes and log rates produced by your Corelight Zeek Sensors. Then adjust your Cortex XSIAM log storage. For more information, see Manage Your Log Storage within Cortex XSIAM.
3. Forward logs to the Syslog Collector.

   Cortex XSIAM can receive logs from Corelight Zeek sensors that use the Syslog export option of RFC5424 over TCP.

   a. In the Syslog configuration of Corelight Zeek (Sensor → Export), specify the details for your Syslog Collector including the hostname or IP address of the Broker VM and corresponding listening port that you defined during activation of the Syslog Collector, default Syslog format (RFC5424), and any log exclusions or filters.

   b. Save your Syslog configuration to apply the configuration to your Corelight Zeek Sensors.

   For full setup instructions, see the Corelight Zeek documentation.

1.6.2.6 | Ingest logs from Fortinet Fortigate firewalls

Abstract

Extend Cortex XSIAM visibility into logs from Fortinet Fortigate firewalls.

If you use Fortinet Fortigate firewalls, you can still take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your firewall logs to Cortex XSIAM . This enables Cortex XSIAM to examine your network traffic to detect anomalous behavior. Cortex XSIAM can use Fortinet Fortigate firewall

logs as the sole data source, but can also use Fortinet Fortigate firewall logs in conjunction with Palo Alto Networks firewall logs. For additional endpoint context, you can also use Cortex XSIAM to collect and alert on endpoint data.

When Cortex XSIAM starts to receive logs, the app can begin stitching network connection logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to generate Analytics issues, and can apply IOC, BIOC, and Correlation Rule matching. You can also use queries to search your network connection logs.

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog collector. You then configure forwarding on your log devices to send logs to the Syslog collector in a CEF format.

1. Verify that your Fortinet Fortigate firewalls meet the following requirements.
    - Must use FortiOS 6.2.1 or a later release
    - `timestamp` must be in nanoseconds
2. Activate the Syslog Collector.
3. Increase log storage for Fortinet Fortigate firewall logs.

    As an estimate for initial sizing, note that the average Fortinet Fortigate log size is roughly 1,070 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Fortinet Fortigate firewalls. For more information, see Manage Your Log Storage within Cortex XSIAM.
4. Configure the log device that receives Fortinet Fortigate firewall logs to forward Syslog events to the Syslog collector in a CEF format.

    Configure your firewall policy to log all traffic and forward the traffic logs to the Syslog collector in a CEF format. By logging all traffic, you enable Cortex XSIAM to detect anomalous behavior from Fortinet Fortigate firewall logs. For more information on setting up Log Forwarding on Fortinet Fortigate firewalls, see the Fortinet FortiOS documentation.

1.6.2.7 | Ingest logs from Microsoft Azure Event Hub

Abstract

Ingest logs from Microsoft Azure Event Hub with an option to ingest audit logs to use in Cortex XSIAM authentication stories.

Cortex XSIAM can ingest different types of data from Microsoft Azure Event Hub using the Microsoft Azure Event Hub data collector. To receive logs from Azure Event Hub, you must configure the Data Sources settings in Cortex XSIAM based on your Microsoft Azure Event Hub configuration. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset (`MSFT_Azure_raw`) that you can use to initiate XQL Search queries. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to normalize Azure Event Hub audit logs, including Azure Kubernetes Service (AKS) audit logs, with other Cortex XSIAM authentication stories across all cloud providers using the same format, which you can query with XQL Search using the `cloud_audit_logs` dataset. For logs that you do not configure Cortex XSIAM to normalize, you can change the default dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from Azure Event Hub logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only raised on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

> **WARNING:**
>
> - Misconfiguration of Event Hub resources could cause ingestion delays.
> - In an existing Event Hub integration, do not change the mapping to a different Event Hub.
> - Do not use the same Event Hub for more than two purposes.

The following table provides a brief description of the different types of Azure audit logs you can collect.

> **NOTE:**
>
> For more information on Azure Event Hub audit logs, see Overview of Azure platform logs.

Type Of Data	Description
Activity logs	Retrieves events related to the operations on each Azure resource in the subscription from the outside in addition to updates on Service Health events. **NOTE:** These logs are from the management plane.
Azure Active Directory (AD) Activity logs and Azure Sign-in logs	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant. **NOTE:** Even though you can collect Azure AD Activity logs and Azure Sign-in logs using the Azure Event Hub data collector, we recommend using the Microsoft Office 365 data collector, because it is easier to configure. In addition, ensure that you do not configure both collectors to collect the same types of logs, because if you do so, you will be creating duplicate data in Cortex XSIAM.
Resource logs, including AKS audit logs	Retrieves events related to operations that were performed within an Azure resource. **NOTE:** These logs are from the data plane.

NOTE:

If you want to ingest raw Microsoft Defender for Endpoint events, use the Microsoft Defender log collector. For more information, see Ingest raw EDR events from Microsoft Defender for Endpoint.

PREREQUISITE:

Ensure that you do the following tasks before you begin configuring data collection from Azure Event Hub.

- Before you set up an Azure Event Hub, calculate the quantity of data that you expect to send to Cortex XSIAM, taking into account potential data spikes and potential increases in data ingestion, because partitions cannot be modified after creation. Use this information to ascertain the optimal number of partitions and Throughput Units (for Azure Basic or Standard) or Processing Units (for Azure Premium). Configure your Event Hub accordingly.
- Create an Azure Event Hub. We recommend using a dedicated Azure Event Hub for this Cortex XSIAM integration. For more information, see Quickstart: Create an event hub using Azure portal.
- Each partition can support a throughput of up to 1 MB/s.
- Ensure the format for the logs you want collected from the Azure Event Hub is either JSON or raw.

Configure the Azure Event Hub collection in Cortex XSIAM:

1. In the Microsoft Azure console, open the Event Hubs page, and select the Azure Event Hub that you created for collection in Cortex XSIAM.
2. Record the following parameters from your configured event hub, which you will need when configuring data collection in Cortex XSIAM.
    - Your event hub's consumer group.
        1. Select Entities → Event Hubs, and select your event hub.
        2. Select Entities → Consumer groups, and select your event hub.
        3. In the Consumer group table, copy the applicable value listed in the Name column for your Cortex XSIAM data collection configuration.
    - Your event hub's connection string for the designated policy.
        1. Select Settings → Shared access policies.
        2. In the Shared access policies table, select the applicable policy.
        3. Copy the Connection string-primary key.
    - Your storage account connection string required for partitions lease management and checkpointing in Cortex XSIAM.
        1. Open the Storage accounts page, and either create a new storage account or select an existing one, which will contain the storage account connection string.
        2. Select Security + networking → Access keys, and click Show keys.
        3. Copy the applicable Connection string.

3. Configure diagnostic settings for the relevant log types you want to collect and then direct these diagnostic settings to the designated Azure Event Hub.

   a. Open the Microsoft Azure console.

   b. Your navigation is dependent on the type of logs you want to configure.

Log Type	**Navigation Path**
Activity logs	Select Azure services → Activity log → Export Activity Logs, and +Add diagnostic setting.
Azure AD Activity logs and Azure Sign-in logs	1. Select Azure services → Azure Active Directory. 2. Select Monitoring → Diagnostic settings, and +Add diagnostic setting.
Resource logs, including AKS audit logs	1. Search for Monitor, and select Settings → Diagnostic settings. 2. From your list of available resources, select the resource that you want to configure for log collection, and then select +Add diagnostic setting. **NOTE:** For every resource that you want to confiure, you'll have to repeat this step, or use Azure policy for a general configuration.

   c. Set the following parameters:

- Diagnostic setting name: Specify a name for your Diagnostic setting.
- Logs Categories/Metrics: The options listed are dependent on the type of logs you want to configure. For Activity logs and Azure AD logs and Azure Sign-in logs, the option is called Logs Categories, and for Resource logs it's called Metrics.

Log Type	Log Categories/Metrics
Activity logs	Select from the list of applicable Activity log categories, the ones that you want to configure your designated resource to collect. We recommend selecting all of the options. ◦ Administrative ◦ Security ◦ ServiceHealth ◦ Alert ◦ Recommendation ◦ Policy ◦ Autoscale ◦ ResourceHealth
Azure AD Activity logs and Azure Sign-in logs	Select from the list of applicable Azure AD Activity and Azure Sign-in Logs Categories, the ones that you want to configure your designated resource to collect. You can select any of the following categories to collect these types of Azure logs. ◦ Azure AD Activity logs: ▪ AuditLogs ◦ Azure Sign-in logs: ▪ SignInLogs ▪ NonInteractiveUserSignInLogs ▪ ServicePrincipalSignInLogs ▪ ManagedIdentitySignInLogs ▪ ADFSSignInLogs **NOTE:** There are additional log categories displayed. We recommend selecting all the available options.
Resource logs, including AKS audit logs	The list displayed is dependent on the resource that you selected. We recommend selecting all the options available for the resource.

- Destination details: Select Stream to event hub, where additional parameters are displayed that you need to configure. Ensure that you set the following parameters using the same settings for the Azure Event Hub that you created for the collection.
  - Subscription: Select the applicable Subscription for the Azure Event Hub.
  - Event hub namespace: Select the applicable Subscription for the Azure Event Hub.
  - (Optional) Event hub name: Specify the name of your Azure Event Hub.
  - Event hub policy: Select the applicable Event hub policy for your Azure Event Hub.

d. Save your settings.

4. Configure the Azure Event Hub collection in Cortex XSIAM.

a. Select Settings → Data Sources.

b. On the Data Sources page, click Add Data Source, search for and select Azure Event Hub, and click Connect.

c. Set these parameters.

- Name: Specify a descriptive name for your log collection configuration.
- Event Hub Connection String: Specify your event hub's connection string for the designated policy.
- Storage Account Connection String: Specify your storage account's connection string for the designated policy.
- Consumer Group: Specify your event hub's consumer group.
- Log Format: Select the log format for the logs collected from the Azure Event Hub as Raw, JSON, CEF, LEEF, Cisco-asa, or Corelight.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the log format is automatically configured. As a result, the Log Format option is removed and is no longer available to configure (default).

  - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

    > **NOTE:**
    >
    > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the Azure Event Hub data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the Azure Event Hub data collector settings. If you did not specify a Vendor or Product in the Azure Event Hub data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

  - Cisco-asa: The following fields are automatically set and not configurable.
    - Vendor: Cisco
    - Product: ASA

    Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.

  - Corelight: The following fields are automatically set and not configurable.
    - Vendor: Corelight
    - Product: Zeek

    Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.

  - Raw or JSON: The following fields are automatically set and are configurable.
    - Vendor: Msft
    - Product: Azure

    Raw or JSON data can be queried in XQL Search using the `msft_azure_raw` dataset.

- Vendor and Product: Specify the Vendor and Product for the type of logs you are ingesting.

  The Vendor and Product are used to define the name of your Cortex Query Language (XQL) dataset (`<vendor>_<product>_raw`). The Vendor and Product values vary depending on the Log Format selected. To uniquely identify the log source, consider changing the values if the values are configurable.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the Vendor and Product fields are automatically configured, so these fields are removed as available options (default).

- Normalize and enrich audit logs: (Optional) For enhanced cloud protection, you can Normalize and enrich audit logs by selecting the checkbox (default). If selected, Cortex XSIAM normalizes and enriches Azure Event Hub audit logs with other Cortex XSIAM authentication stories across all cloud providers using the same format. You can query this normalized data with XQL Search using the `cloud_audit_logs` dataset.

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears underneath the Azure Event Hub configuration with the amount of data received.

1.6.2.8 | Ingest network flow logs from Microsoft Azure Network Watcher

Abstract

Ingest network security group (NSG) or Virtual network (VNet) flow logs from Microsoft Azure Network Watcher for use in Cortex XSIAM network stories.

To receive network security group (NSG) or Virtual network (VNet) flow logs from Azure Network Watcher, you must configure data collection from Microsoft Azure Network Watcher using an Azure Function provided by Cortex XSIAM. This Azure Function requires a token that is generated when you configure your Azure Network Watcher Collector in Cortex XSIAM. After you have configured the Cortex XSIAM collector and successfully deployed the Azure Function to your Azure account, Cortex XSIAM will start receiving and ingesting network flow logs from Azure Network Watcher.

The Azure Network Watcher Collector is deployed using an ARM template. During deployment, the template retrieves keys using the `listKeys` function, and your app can bind to the blob storage using the connection string generated from those keys. After deployment, this binding works without the need to provide any connection string manually, because the keys were already retrieved and injected during deployment.

In addition to the user-specified storage account that captures the log blobs, the template also creates a secondary, internal storage account for internal operations related to the function app. This internal storage account is used by the function app for operations such as storing function state, and intermediate processing. To enhance security, public network access is disabled, and the account is restricted to private endpoints only. This additional internal storage account allows the function app to securely store data without relying on the user-specified storage account for internal processes. This separation enhances data security and isolation between user-facing storage and internal application operations. VNet integration is required only for the internal storage account's internal operations. The user-specified storage account used for NSG or VNet flow logs does not require VNet integration.

When Cortex XSIAM begins receiving logs, the app creates a new dataset (`MSFT_Azure_raw`) that you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to ingest network flow logs as Cortex XSIAM network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC) when relevant from Azure Network Watcher flow logs. While Correlation Rules issues are raised on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

PREREQUISITE:

- For NSG:
  - Ensure that your NSG flow logs in Azure Network Watcher conform to the requirements as outlined in Microsoft documentation. For more information, see Introduction to flow logging for network security groups.
  - Enable NSG flow logs in the Microsoft Azure Portal.
- For VNet:
  - Ensure that your VNet flow logs in Azure Network Watcher conform to the requirements as outlined in Microsoft documentation. For more information, see Introduction to flow logging for virtual networks.
  - Enable VNet flow logs in the Microsoft Azure Portal.
- Ensure that you have an Azure subscription with user role permissions to deploy ARM templates and create the required resources.

  The `listKeys` function in an Azure Resource Manager (ARM) template retrieves the storage account keys, and it requires special permissions to execute. Specifically, the user or identity running the ARM template needs the following permission: `Microsoft.Storage/storageAccounts/listKeys/action`. If the user or service principal running the ARM template has the necessary user role (such as Owner or Storage Account Contributor), permission is implicitly granted for the template to retrieve the storage account keys.
- Perform this procedure in the order shown below, because you need to save a token and a URL from Cortex XSIAM in earlier steps, and use them in Azure in later steps.

1. Configure the Azure Network Watcher collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Azure Network Watcher, and click Connect.

   c. Set these parameters:

   - Name: Specify a meaningful name for your log collection configuration.
   - Enhanced Cloud Protection: (Optional) For enhanced cloud protection, you can normalize and enrich flow logs by selecting the Use flow logs in analytics checkbox. If selected, Cortex XSIAM ingests network flow logs as Cortex XSIAM network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`.

   d. Click Save & Generate Token. The token is displayed in a popup.

   Click the copy icon next to the key and save the copy of this token somewhere safe. You will need to provide this token when you configure the Azure Function and set the Cortex Access Token value. If you forget to record the token and close the window, you will need to generate a new one and repeat this process. When you are finished, click Done to close the window.

e. On the Integrations page for the Azure Network Watch Collector that you created, click the Copy API URL icon and save a copy of the URL somewhere safe. You will need to provide this URL when you configure the Azure Function and set the Cortex Http Endpoint value.

2. Configure the Azure Function provided by Cortex XSIAM.

a. Do one of the following, depending on the flow log type:

- For NSG, open this Azure Function provided by Cortex XSIAM.
- For VNet, open this Azure Function provided by Cortex XSIAM.

b. Click Deploy to Azure.

c. Log in to Azure, and if necessary, complete authentication procedures.

d. Set these parameters, where some fields are mandatory to set and others may already be populated for you.

- Subscription: Specify the Azure subscription that you want to use for the App Configuration. If your account has only one subscription, it is automatically selected.
- Resource group: Specify or create a resource group for your App Configuration store resource.
- Region: Specify the Azure region that you want to use.
- Unique Name: Enter a unique name for the function app. The name that you provide will be concatenated to some of the resource names, to make it easier to locate the related resources later on. The name must only contain alphanumeric characters (letters and numbers, no special symbols) and must contain no more than 10 characters.
- Cortex Access Token: Cortex HTTP authorization key that you recorded when you configured the Azure Network Watcher collection in Cortex XSIAM in an earlier step.
- Target Storage Account Name: Enter the name of the Azure Storage Account that was created during the NSG or VNet flow logs setup in Azure Network Watcher, where the log blobs are being stored.
- Target Container Name: This field should be left empty for most use cases.

  For NSG, the default value `insights-logs-networksecuritygroupflowevent` is the name that is automatically created for the container during configuration of the network watcher.

  For VNet, the default value `insights-logs-flowlogflowevent` is the name that is automatically created for the container during configuration of the network watcher.
- Location: The region where all the resources will be deployed (leave blank to use the same region as the resource group).
- Cortex Http Endpoint: Specify the API URL that you recorded when you configured the Azure Network Watcher collection in Cortex XSIAM.
- Remote Package: The URL of the remote package ZIP file containing the Azure Function code. Keep the default value, unless instructed otherwise.

e. Click Review + Create to confirm your settings for the Azure Function.

f. Click Create. It can take a few minutes until the deployment is complete.

> **NOTE:**
>
> In addition to your storage account, the template automatically creates another storage account that is required by the function app for internal use only. The internal storage account name is prefixed with `cortex` and is followed by a unique suffix based on the resource group, storage account, and container names.

After events start to come in, a green check mark appears underneath the Azure Network Watcher configuration that you created in Cortex XSIAM, and the amount of data received is displayed.

1.6.2.9 | Ingest logs and data from Okta

Abstract

Ingest authentication logs and data from Okta for use in Cortex XSIAM authentication stories.

To receive logs and data from Okta, you must configure the Data Sources settings in Cortex XSIAM. After you set up data collection, Cortex XSIAM immediately begins receiving new logs and data from the source. The information from Okta is then searchable in XQL Search using the `okta_sso_raw` dataset. In addition, depending on the event type, data is normalized to either `xdr_data` or `saas_audit_logs` datasets.

You can collect all types of events from Okta. When setting up the Okta data collector in Cortex XSIAM, a field called Okta Filter is available to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq “user.session.start”.\n`. For Okta information to be woven into authentication stories, “`user.authentication.sso`” events must be collected.

The Okta API enforces concurrent rate limits. The Okta data collector is built with a mechanism which reduces the amount of requests whenever an error is received from the Okta API indicating that too many requests have already been sent. In addition, to ensure you are properly notified about this, an alert is displayed in the Notification Area and a record is added to the Management Audit Logs.

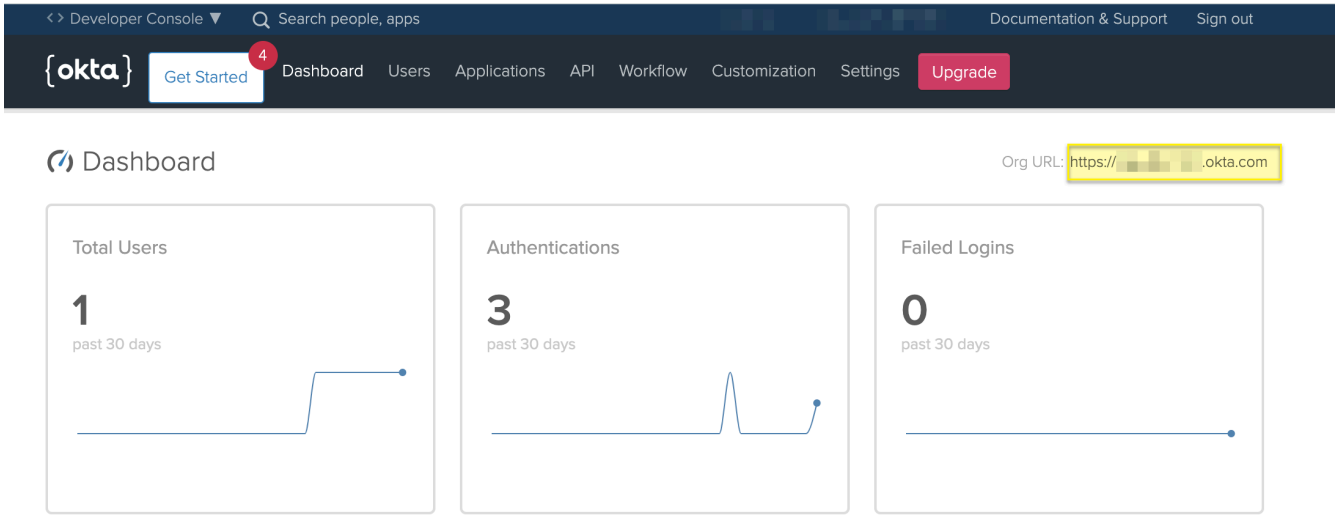
Before you begin configuring data collection from Okta, ensure your Okta user has administrator privileges with a role that can create API tokens, such as the read-only administrator, Super administrator, and Organization administrator. For more information, see the Okta Administrators Documentation.

To configure the Okta collection in Cortex XSIAM:

1. Identify the domain name of your Okta service.

   From the Dashboard of your Okta console, note your Org URL.

   For more information, see the Okta Documentation.



2. Obtain your authentication token in Okta.

   a. Select API → Tokens.

   b. Create Token and record the token value.

      This is your only opportunity to record the value.

3. Select Settings → Data Sources.

4. On the Data Sources page, click Add Data Source, search for and select Okta, and click Connect.

5. Integrate the Okta authentication service with Cortex XSIAM.

   a. Specify the OKTA DOMAIN (Org URL) that you identified on your Okta console.

   b. Specify the TOKEN used to authenticate with Okta.

   c. Specify the Okta Filter to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq “user.session.start”.\n`. For Okta information to be weaved into authentication stories, “`user.authentication.sso`” events must be collected.

   d. Test the connection settings.

   e. If successful, Enable Okta log collection.

      Once events start to come in, a green check mark appears underneath the Okta configuration with the amount of data received.

6. After Cortex XSIAM begins receiving information from the service, you can Create an XQL Query to search for specific data. When including authentication events, you can also Create an Authentication Query to search for specific authentication data.

1.6.2.10 | Ingest logs from Windows DHCP using Elasticsearch Filebeat

Abstract

Learn how to configure Cortex XSIAM to receive Windows DHCP logs.

You can configure Cortex XSIAM to receive Windows DHCP logs using Elasticsearch Filebeat with the following data collectors.

Ingest Windows DHCP logs with an XDR Collector profile

Abstract

Extend Cortex XSIAM visibility into logs from Windows DHCP using an XDR Collector Windows Filebeat profile.

Extend Cortex XSIAM visibility into logs from Windows DHCP using an XDR Collector Windows Filebeat profile.

You can enrich network logs with Windows DHCP data when defining data collection in an XDR Collector Windows Filebeat profile. When you add a XDR Collector Windows Filebeat profile using the Elasticsearch Filebeat default configuration file called `filebeat.yml`, you can define whether the collected data undergoes follow-up processing in the backend for Windows DHCP data. Cortex XSIAM uses Windows DHCP logs to enrich your network logs with hostnames and MAC addresses that are searchable in XQL Search using the Windows DHCP Cortex Query Language (XQL) dataset (`microsoft_dhcp_raw`).

While this enrichment is also available when configuring a Windows DHCP Collector for a cloud data collection integration, we recommend configuring Cortex XSIAM to receive Windows DHCP logs with an XDR Collector Windows Filebeat profile because it's the ideal setup configuration.

Configure Cortex XSIAM to receive logs from Windows DHCP using an XDR Collector Windows Filebeat profile.

1. Add an XDR Collector Profile for Windows.

   Follow the steps for creating a Windows Filebeat profile as described in Add an XDR Collector Profile for Windows, and in the Filebeat Configuration File area, ensure that you select and Add the DHCP template. The template's content will be displayed here, and is editable.

2. To configure collection of Windows DHCP data, edit the template text as necessary for your system.

   You can enrich network logs with Windows DHCP data when defining data collection by setting the `vendor` to **"**`microsoft`**"** , and `product` to **"**`dhcp`**"** in the `filebeat.yml` file, which you can then query in the `microsoft_dhcp_raw` dataset.

   > **NOTE:**
   >
   > To avoid formatting issues in `filebeat.yml`, we recommend that you edit the text file inside the user interface, instead of copying it and editing it elsewhere. Validate the syntax of the YML file before you finish creating the profile.

Ingest Windows DHCP logs with the Windows DHCP Collector

Abstract

Extend Cortex XSIAM visibility into logs from Windows DHCP using Elasticsearch Filebeat with the Windows DHCP data collector.

Extend Cortex XSIAM visibility into logs from Windows DHCP using Elasticsearch Filebeat with the Windows DHCP data collector.

To receive Windows DHCP logs, you must configure data collection from Windows DHCP via Elasticsearch Filebeat. This is configured by setting up a Windows DHCP Collector in Cortex XSIAM and installing and configuring an Elasticsearch Filebeat agent on your Windows DHCP Server. Cortex XSIAM supports using Filebeat up to version 8.0.1 with the Windows DHCP Collector.

Certain settings in the Elasticsearch Filebeat default configuration file called `filebeat.yml` must be populated with values provided when you configure the Data Sources settings in Cortex XSIAM for the Windows DHCP Collector. To help you configure the `filebeat.yml` correctly, Cortex XSIAM provides an example file that you can download and customize. After you set up collection integration, Cortex XSIAM begins receiving new logs and data from the source.

> **NOTE:**
>
> For more information on configuring the `filebeat.yml` file, see the Elastic Filebeat Documentation.

Windows DHCP logs are stored as CSV (comma-separated values) log files. The logs rotate by days (`DhcpSrvLog-<day>.log`), and each file contains two sections: `Event ID Meaning` and the events list.

As soon as Cortex XSIAM begins receiving logs, the app automatically creates a Windows DHCP XQL dataset (`microsoft_dhcp_raw`). Cortex XSIAM uses Windows DHCP logs to enrich your network logs with hostnames and MAC addresses that are searchable in XQL Search using the Windows DHCP Cortex Query Language (XQL) dataset.

Configure Cortex XSIAM to receive logs from Windows DHCP via Elasticsearch Filebeat with the Windows DHCP collector.

1. Configure the Windows DHCP Collector in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Windows DHCP, and click Connect.

   c. (Optional) Download example filebeat.yml file.

   To help you configure your `filebeat.yml` file correctly, Cortex XSIAM provides an example `filebeat.yml` file that you can download and customize. To download this file, use the link provided in this dialog box.

   > **NOTE:**
   >
   > To avoid formatting issues in your `filebeat.yml`, we recommend that you use the download example file to make your customizations. Do not copy and paste the code syntax examples provided later in this procedure into your file.

d. Specify a descriptive Name for your log collection configuration.

e. Save & Generate Token. The token is displayed in a blue box, which is blurred out in the image below.

Click the copy icon next to the key and record it somewhere safe. You will need to provide this key when you set the `api_key` value in the Elasticsearch Output section in the `filebeat.yml` file as explained in Step #2. If you forget to record the key and close the window you will need to generate a new key and repeat this process.

f. Select Done to close the window.

g. In the Integrations page for the Windows DHCP Collector that you created, select Copy api url and record it somewhere safe. You will need to provide this URL when you set the `hosts` value in the Elasticsearch Output section in the `filebeat.yml` file as explained in Step #2.

2. Configure an Elasticsearch Filebeat agent on your Windows DHCP Server.

a. Navigate to the Elasticsearch Filebeat installation directory, and open the `filebeat.yml` file to configure data collection with Cortex XSIAM. We recommend that you use the download example file provided by Cortex XSIAM.

b. Update the following sections and tags in the `filebeat.yml` file. The example code below details the specific sections to make these changes in the file.

- Filebeat inputs: Define the paths to crawl and fetch. The code below provides an example of how to configure the Filebeat inputs section in the `filebeat.yml` file with these paths configured.

```
# ============================== Filebeat inputs ===============================
filebeat.inputs:
  # Each - is an input. Most options can be set at the input level, so
  # you can use different inputs for various configurations.
  # Below are the input specific configurations.
  - type: log
    # Change to true to enable this input configuration.
    enabled: true
    # Paths that should be crawled and fetched. Glob based paths.
    paths:
      - c:\Windows\System32\dhcp\DhcpSrvLog*.log
```

- Elasticsearch Output: Set the `hosts` and `api_key`, where both of these values are obtained when you configured the Windows DHCP Collector in Cortex XSIAM as explained in Step #1. The code below provides an example of how to configure the Elasticsearch Output section in the `filebeat.yml` file and indicates which settings need to be obtained from Cortex XSIAM.

```
# ---------------------------- Elasticsearch Output ----------------------------
output.elasticsearch:
  enabled: true
  # Array of hosts to connect to.
  hosts: ["OBTAIN THIS URL FROM CORTEX XDR"]
  # Protocol - either `http` (default) or `https`.
  protocol: "https"
  compression_level: 5
  # Authentication credentials - either API key or username/password.
  api_key: "OBTAIN THIS KEY FROM CORTEX XDR"
```

- Processors: Set the `tokenizer` and add a `drop_event processor` to drop all events that do not start with an event ID. The code below provides an example of how to configure the Processors section in the `filebeat.yml` file and indicates which settings need to be obtained from Cortex XSIAM.

  > **NOTE:**
  >
  > The `tokenizer` definition is dependent on the Windows server version that you are using as the log format differs.
  >
  > -For platforms earlier than Windows Server 2008, use `"%{id},%{date},%{time},%{description},%{ipAddress},%{hostName},%{macAddress}"`
  >
  > -For Windows Server 2008 and 2008 R2, use `"%{id},%{date},%{time},%{description},%{ipAddress},%{hostName},%{macAddress},%{userName},%{transactionID},%{qResult},%{probationTime},%{correlationID}"`
  >
  > For Windows Server 2012 and above, use `"%{id},%{date},%{time},%{description},%{ipAddress},%{hostName},%{macAddress},%{userName},%{transactionID},%{qResult},%{probationTime},%{correlationID},%{dhcid},%{vendorClassHex},%{vendorClassASCII},%{userClassHex},%{userClassASCII},%{relayAgentInformation},%{dnsRegError}"`

```
# ================================ Processors ================================
processors:
  - add_host_metadata:
    when.not.contains.tags: forwarded
  - drop_event.when.not.regexp.message: "^[0-9]+,.*"
  - dissect:
    tokenizer: "%{id},%{date},%{time},%{description},%{ipAddress},%{hostName},%{macAddress},%{userName},%{transactionID},%{qResult},%{probationTime},%{correlationID},%{dhcid},%{vendorClassHex},%{vendorClassASCII},%{userClassHex},%{userClassASCII},%{relayAgentInformation},%{dnsRegError}"
  - drop_fields:
    fields: ["message"]
  - add_locale: ~
  - rename:
      fields:
        - from: "event.timezone"
          to: "dissect.timezone"
      ignore_missing: true
      fail_on_error: false
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~
```

3. Verify the status of the integration.

   Return to the Integrations page and view the statistics for the log collection configuration.

4. After Cortex XSIAM begins receiving logs from Windows DHCP via Elasticsearch Filebeat, you can use the XQL Search to search for logs in the new dataset (`microsoft_dhcp_raw`).

1.6.2.11 | Ingest logs from Zscaler Internet Access

Abstract

Extend Cortex XSIAM visibility into logs from Zscaler Internet Access (ZIA).

If you use Zscaler Internet Access (ZIA) in your network, you can forward your firewall and network logs to Cortex XSIAM for analysis. This enables you to take advantage of Cortex XSIAM anomalous behavior detection and investigation capabilities. Cortex XSIAM can use the firewall and network logs from ZIA as the sole data source, and can also use these firewall and network logs from ZIA in conjunction with Palo Alto Networks firewall and network logs. For additional endpoint context, you can also use Cortex XSIAM to collect and alert on endpoint data.

To integrate your logs, you first need to set up an applet in a broker VM within your network to act as a Syslog Collector. You then configure forwarding on your log devices to send logs to the Syslog collector in a CEF format. To provide seamless log ingestion, Cortex XSIAM automatically maps the fields in your traffic logs to the Cortex XSIAM log format.

When Cortex XSIAM starts to receive logs, the app performs these actions.

- Begins stitching network connection and firewall logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to generate Analytics issues and can apply IOC, BIOC, and Correlation Rule matching. You can also use queries to search your network connection logs.
- Creates a Zscaler Cortex Query Language (XQL) dataset, which enables you to search the logs using XQL Search. The Zscaler XQL datasets are dependent on the ZIA NSS Feed that you've configured for the types of logs you want to collect.
    - Firewall logs: `zscaler_nssfwlog_raw`
    - Web logs: `zscalar_nssweblog_raw`

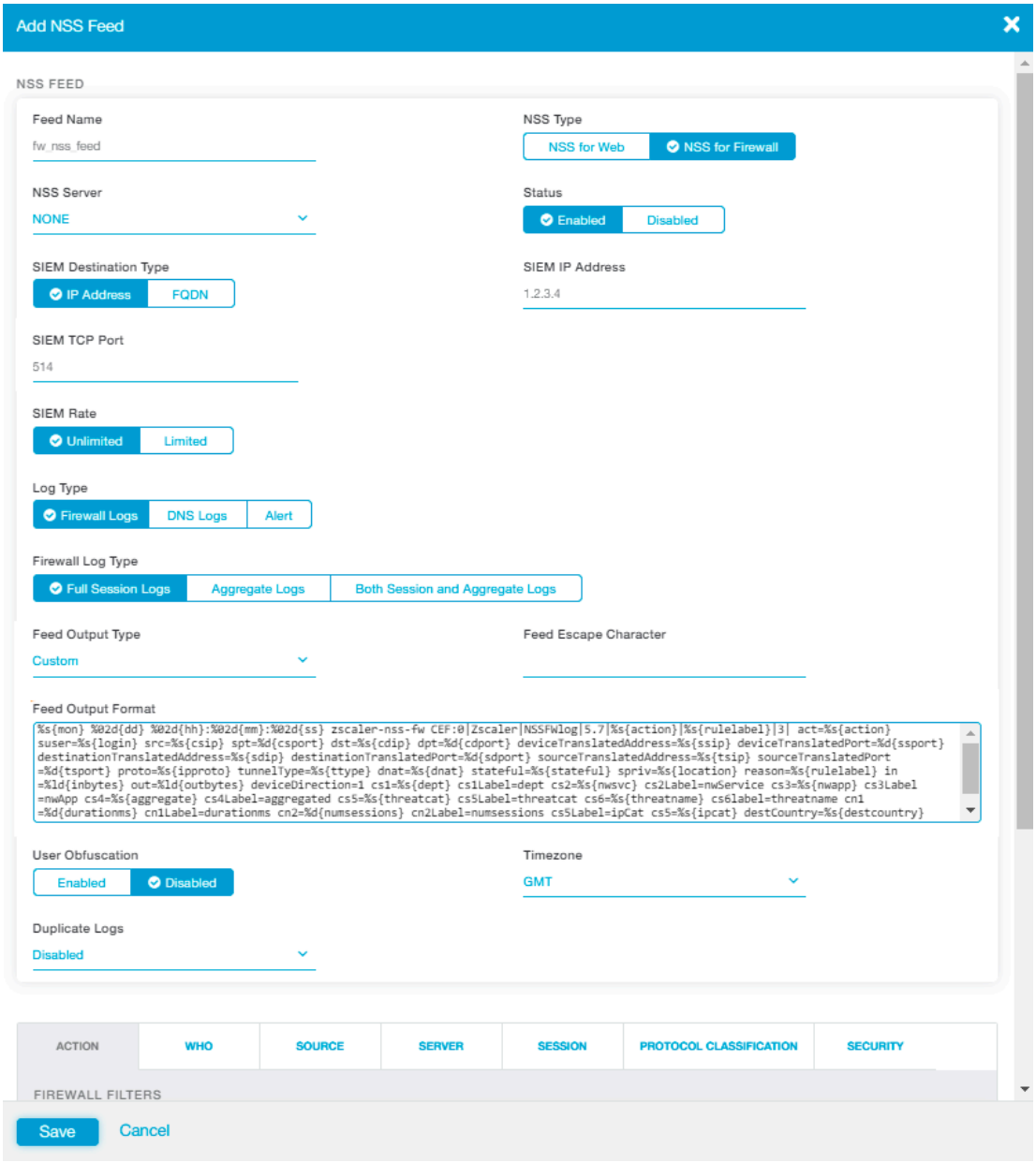
To ingest logs from Zscaler Internet Access (ZIA):

1. Activate the Syslog Collector.
2. Increase log storage for ZIA logs. For more information, see Manage Your Log Storage.
3. Configure NSS log forwarding in Zscaler Internet Access to the Syslog Collector in a CEF format.
    a. In the Zscaler Internet Access application, select Administration → Nanolog Streaming Service.
    b. In the NSS Feeds tab, Add NSS Feed.
    c. In the Add NSS Feed screen, configure the fields for the Cortex XSIAM Syslog Collector.

    The steps below differ depending on the type of NSS Feed you are configuring to collect either firewall logs or web logs. For more information on all the configurations available on the screen, see the ZIA documentation:

    - Firewall logs: See Adding NSS Feeds for Firewall Logs.
    - Web logs: See Adding NSS Feeds for Web Logs.

    The following image displays the fields required to add an NSS feed.


Add NSS Feed
NSS FEED
Feed Name
fw_nss_feed
NSS Type
NSS for Web
NSS for Firewall
NSS Server
NONE
Status
Enabled
Disabled
SIEM Destination Type
IP Address
FQDN
SIEM IP Address
1.2.3.4
SIEM TCP Port
514
SIEM Rate
Unlimited
Limited
Log Type
Firewall Logs
DNS Logs
Alert
Firewall Log Type
Full Session Logs
Aggregate Logs
Both Session and Aggregate Logs
Feed Output Type
Custom
Feed Escape Character
Feed Output Format
%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-fw CEF:0|Zscaler|NSSFWlog|5.7|%s{action}|%s{rulelabel}|3| act=%s{action}
suser=%s{login} src=%s{csip} spt=%d{csport} dst=%s{cdip} dpt=%d{cdport} deviceTranslatedAddress=%s{ssip} deviceTranslatedPort=%d{ssport}
destinationTranslatedAddress=%s{sdip} destinationTranslatedPort=%d{sdport} sourceTranslatedAddress=%s{tsip} sourceTranslatedPort
=%d{tsport} proto=%s{ipproto} tunnelType=%s{ttype} dnat=%s{dnat} stateful=%s{stateful} spriv=%s{location} reason=%s{rulelabel} in
=%ld{inbytes} out=%ld{outbytes} deviceDirection=1 cs1=%s{dept} cs1Label=dept cs2=%s{nwsvc} cs2Label=nwService cs3=%s{nwapp} cs3Label
=nwApp cs4=%s{aggregate} cs4Label=aggregated cs5=%s{threatcat} cs5Label=threatcat cs6=%s{threatname} cs6label=threatname cn1
=%d{durationms} cn1Label=durationms cn2=%d{numsessions} cn2Label=numsessions cs5Label=ipCat cs5=%s{ipcat} destCountry=%s{destcountry}
User Obfuscation
Enabled
Disabled
Timezone
GMT
Duplicate Logs
Disabled
ACTION
WHO
SOURCE
SERVER
SESSION
PROTOCOL CLASSIFICATION
SECURITY
FIREWALL FILTERS
Save
Cancel

- NSS Type: Select either NSS for Web (default) to collect web logs or NSS for Firewall to collect firewall logs.
- SIEM TCP Port: Specify the port that you set when activating the Syslog Collector in Cortex XSIAM. See Activate the Syslog Collector.
- SIEM IP Address: Specify the IP that you set when activating the Syslog Collector in Cortex XSIAM. See Activate the Syslog Collector.
- Feed Escape Character: Specify the feed escape character as =.
- Feed Output Type: Select Custom.
- Feed Output Format: Specify the output format, which is dependent on the type of logs you are collecting as defined in the NSS Type field:

Log Type	Feed Output Format
Firewall logs	`%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss-fw CEF:0|Zscaler|NSSFWlog|5.7|%s{action}|%s{rulelabel}|3|act=%s{action} suser=%s{login} src=%s{csip} spt=%d{csport} dst=%s{cdip} dpt=%d{cdport} deviceTranslatedAddress=%s{ssip} deviceTranslatedPort=%d{ssport} destinationTranslatedAddress=%s{sdip} destinationTranslatedPort=%d{sdport} sourceTranslatedAddress=%s{tsip} sourceTranslatedPort=%d{tsport} proto=%s{ipproto} tunnelType=%s{ttype} dnat=%s{dnat} stateful=%s{stateful} spriv=%s{location} reason=%s{rulelabel} in=%ld{inbytes} out=%ld{outbytes} rt=%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} deviceDirection=1 cs1=%s{dept} cs1Label=dept cs2=%s{nwsvc} cs2Label=nwService cs3=%s{nwapp} cs3Label=nwApp cs4=%s{aggregate} cs4Label=aggregated cs6=%s{threatname} cs6label=threatname cn1=%d{durationms} cn1Label=durationms cn2=%d{numsessions} cn2Label=numsessions cs5Label=ipCat cs5=%s{ipcat} cat=%s{threatcat} destCountry=%s{destcountry} avgduration=%d{avgduration}`
Web logs	`%s{mon} %02d{dd} %02d{hh}:%02d{mm}:%02d{ss} zscaler-nss CEF:0|Zscaler|NSSWeblog|5.0|%s{action}|%s{reason}|3|act=%s{action} app=%s{proto} cat=%s{urlcat} dhost=%s{ehost} dst=%s{sip} src=%s{cip} in=%d{respsize} outcome=%s{respcode} out=%d{reqsize} request=%s{eurl} rt=%s{mon} %02d{dd} %d{yy} %02d{hh}:%02d{mm}:%02d{ss} sourceTranslatedAddress=%s{cintip} requestClientApplication=%s{ua} requestMethod=%s{reqmethod} suser=%s{login} spriv=%s{location} externalId=%d{recordid} fileType=%s{filetype} reason=%s{reason} destinationServiceName=%s{appname} cn1=%d{riskscore} cn1Label=riskscore cs1=%s{dept} cs1Label=dept cs2=%s{urlsupercat} cs2Label=urlsupercat cs3=%s{appclass} cs3Label=appclass cs4=%s{malwarecat} cs4Label=malwarecat cs5=%s{threatname} cs5Label=threatname cs6=%s{dlpeng} cs6Label=dlpeng ZscalerNSSWeblogURLClass=%s{urlclass} ZscalerNSSWeblogDLPDictionaries=%s{dlpdict} requestContext=%s{ereferer} contenttype=%s{contenttype} unscannabletype=%s{unscannabletype} deviceowner=%s{deviceowner} devicehostname=%s{devicehostname}\n`

d. Click Save.

e. Click Save and activate the change according to the Zscaler Internet Access (ZIA) documentation.

1.6.2.12 | Ingest logs from Zscaler Private Access

Abstract

Extend Cortex XSIAM visibility into logs from Zscaler Private Access (ZPA).

If you use Zscaler Private Access (ZPA) in your network as an alternative to VPNs, you can forward your network logs to Cortex XSIAM for analysis. This enables you to take advantage of Cortex XSIAM anomalous behavior detection and investigation capabilities. Cortex XSIAM can use the network logs from ZPA as the sole data source, and can also use these network logs from ZPA in conjunction with Palo Alto Networks network logs.

When Cortex XSIAM starts to receive logs, the following actions are performed:

- Stitching network connection logs with other logs to form network stories. Cortex XSIAM can also analyze your logs to apply IOC, BIOC, and Correlation Rules matching. You can also use queries to search your network connection logs.
- Creates a Zscaler Cortex Query Language (XQL) dataset (`zscaler_zpa_raw`), which enables you to search the logs using XQL Search.

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog Collector. You then configure forwarding on your log devices to send logs to the Syslog collector in a LEEF format. To provide seamless log ingestion, Cortex XSIAM automatically maps the fields in your traffic logs to the Cortex XSIAM log format.

Prerequisite Step

Before you can add a log receiver in Zscaler Private Access, as explained in the task below, you must first deploy your App Connectors. For more information, see App Connector Deployment Guides for Supported Platforms.

To ingest logs from Zscaler Private Access (ZPA):

1. Activate the Syslog Collector.
2. Increase log storage for ZPA logs. For more information, see Manage Your Log Storage.
3. Configure ZPA log forwarding in Zscaler Private Access to the Syslog Collector in a LEEF format.
    a. In the Zscaler Private Access application, select Administration → Log Receivers.
    b. Click Add Log Receiver.

    > **NOTE:**
    >
    > For more information on configuring the parameters on the screen, see the Zscaler Private Access (ZPA) documentation for Configuring a Log Receiver.

    c. In the Add Log Receiver window, configure the following fields on the Log Receiver tab:

    - Name: Specify a name for the log receiver. The name cannot contain special characters, with the exception of periods (.), hyphens (-), and underscores (_).
    - Description: (Optional) Specify a log receiver description.
    - Domain or IP Address: Specify the fully qualified domain name (FQDN) or IP address for the log receiver that you set when activating the Syslog Collector in Cortex XSIAM. See Activate Syslog Collector.
    - TCP Port: Specify the TCP port number used by the log receiver that you set when activating the Syslog Collector in Cortex XSIAM. See Activate Syslog Collector.
    - TLS Encryption: Toggle to Enabled to encrypt traffic between the log receiver and your Syslog Collector in Cortex XSIAMusing mutually authenticated TLS communication. To use this setting, the log receiver must support TLS communication. For more information, see About the Log Streaming Service.
    - App Connector Groups: (Optional) Select the App Connector groups that can forward logs to the receiver, and click Done. You can search for a specific group, click Select All to apply all groups, or click Clear Selection to remove all selections.

    d. Click Next.
    e. Configure the following fields in the Log Stream tab:

- Log Type: Select the log type you want to collect, where only the following logs types are currently supported to collect with your Syslog Collector in Cortex XSIAM:

  > **NOTE:**
  >
  > You can only configure a ZPA log receiver to collect one type of log with your Syslog Collector in Cortex XSIAM. To configure more that one log type, you'll need to add another log receiver.

  - User Activity: Information on end user requests to applications. For more information, see User Activity Log Fields.
  - User Status: Information related to an end user's availability and connection to ZPA. For more information, see User Status Log Fields.
  - App Connector Status: Information related to an App Connector's availability and connection to ZPA. For more information, see About App Connector Status Log Fields.
  - Audit Logs: Session information for all admins accessing the ZPA Admin Portal. For more information, See About Audit Log Fields and About Audit Logs.
- Log Template: Select a Custom template.
- Log Stream Content: Create the log template that you require, according to the Log Type you've selected, using the Zscaler documentation mentioned in previous steps as a reference.

  If you copy and modify the following examples in the table below, validate your log template using an editor, ensuring that there are no additional spaces or line breaks, and then copy and paste it into the Log Stream Content field.

Log Type	Log Template
User activity	`LEEF:1.0|Zscaler|ZPA|4.1|%s{ConnectionStatus}%s{InternalReason}|cat=ZPA User Activity\tdevTime=%s{LogTimestamp:epoch}\tCustomer=%s{Customer}\tSessionID=%s{SessionID}\tConnectionID=%s{ConnectionID}\tInternalReason=%s{InternalReason}\tConnectionStatus=%s{ConnectionStatus}\tproto=%d{IPProtocol}\tDoubleEncryption=%d{DoubleEncryption}\tusrName=%s{Username}\tdstPort=%d{ServicePort}\tsrc=%s{ClientPublicIP}\tsrcPreNAT=%s{ClientPrivateIP}\tClientLatitude=%f{ClientLatitude}\tClientLongitude=%f{ClientLongitude}\tClientCountryCode=%s{ClientCountryCode}\tClientZEN=%s{ClientZEN}\tpolicy=%s{Policy}\tConnector=%s{Connector}\tConnectorZEN=%s{ConnectorZEN}\tConnectorIP=%s{ConnectorIP}\tConnectorPort=%d{ConnectorPort}\tApplicationName=%s{Host}\tApplicationSegment=%s{Application}\tAppGroup=%s{AppGroup}\tServer=%s{Server}\tdst=%s{ServerIP}\tServerPort=%d{ServerPort}\tPolicyProcessingTime=%d{PolicyProcessingTime}\tServerSetupTime=%d{ServerSetupTime}\tTimestampConnectionStart:iso8601=%s{TimestampConnectionStart:iso8601}\tTimestampConnectionEnd:iso8601=%s{TimestampConnectionEnd:iso8601}\tTimestampCATx:iso8601=%s{TimestampCATx:iso8601}\tTimestampCARx:iso8601=%s{TimestampCARx:iso8601}\tTimestampAppLearnStart:iso8601=%s{TimestampAppLearnStart:iso8601}\tTimestampZENFirstRxClient:iso8601=%s{TimestampZENFirstRxClient:iso8601}\tTimestampZENFirstTxClient:iso8601=%s{TimestampZENFirstTxClient:iso8601}\tTimestampZENLastRxClient:iso8601=%s{TimestampZENLastRxClient:iso8601}\tTimestampZENLastTxClient:iso8601=%s{TimestampZENLastTxClient:iso8601}\tTimestampConnectorZENSetupComplete:iso8601=%s{TimestampConnectorZENSetupComplete:iso8601}\tTimestampZENFirstRxConnector:iso8601=%s{TimestampZENFirstRxConnector:iso8601}\tTimestampZENFirstTxConnector:iso8601=%s{TimestampZENFirstTxConnector:iso8601}\tTimestampZENLastRxConnector:iso8601=%s{TimestampZENLastRxConnector:iso8601}\tTimestampZENLastTxConnector:iso8601=%s{TimestampZENLastTxConnector:iso8601}\tZENTotalBytesRxClient=%d{ZENTotalBytesRxClient}\tZENBytesRxClient=%d{ZENBytesRxClient}\tZENTotalBytesTxClient=%d{ZENTotalBytesTxClient}\tZENBytesTxClient=%d{ZENBytesTxClient}\tZENTotalBytesRxConnector=%d{ZENTotalBytesRxConnector}\tZENBytesRxConnector=%d{ZENBytesRxConnector}\tZENTotalBytesTxConnector=%d{ZENTotalBytesTxConnector}\tZENBytesTxConnector=%d{ZENBytesTxConnector}\tIdp=%s{Idp}\n`
User status	`LEEF:1.0|Zscaler|ZPA|4.1|%s{SessionStatus}|cat=ZPA User Status\tdevTime=%s{LogTimestamp:epoch}\tCustomer=%s{Customer}\tusrName=%s{Username}\tSessionID=%s{SessionID}\tSessionStatus=%s{SessionStatus}\tVersion=%s{Version}\tZEN=%s{ZEN}\tCertificateCN=%s{CertificateCN}\tsrcPreNAT=%s{PrivateIP}\tsrc=%s{PublicIP}\tLatitude=%f{Latitude}\tLongitude=%f{Longitude}\tCountryCode=%s{CountryCode}\tTimestampAuthentication:iso8601=%s{TimestampAuthentication:iso8601}\tTimestampUnAuthentication:iso8601=%s{TimestampUnAuthentication:iso8601}\tdstBytes=%d{TotalBytesRx}\tsrcBytes=%d{TotalBytesTx}\tIdp=%s{Idp}\tidentHostName=%s{Hostname}\tPlatform=%s{Platform}\tClientType=%s{ClientType}\tTrustedNetworks=%s(,){TrustedNetworks}\tTrustedNetworksNames=%s(,){TrustedNetworksNames}\tSAMLAttributes=%s{SAMLAttributes}\tPosturesHit=%s(,){PosturesHit}\tPosturesMiss=%s(,){PosturesMiss}\tZENLatitude=%f{ZENLatitude}\tZENLongitude=%f{ZENLongitude}\tZENCountryCode=%s{ZENCountryCode}\n`

Log Type	Log Template
App connector status	LEEF:1.0|Zscaler|ZPA|4.1|%s{SessionStatus}|cat=Connector Status \tdevTime=%s{LogTimestamp:epoch}\tCustomer=%s{Customer}\tSessionID=%s{SessionID} \tSessionType=%s{SessionType}\tVersion=%s{Version}\tPlatform=%s{Platform} \tZEN=%s{ZEN}\tConnector=%s{Connector}\tConnectorGroup=%s{ConnectorGroup} \tsrcPreNAT=%s{PrivateIP}\tsrc=%s{PublicIP}\tLatitude=%f{Latitude} \tLongitude=%f{Longitude}\tCountryCode=%s{CountryCode} \tTimestampAuthentication:iso8601=%s{TimestampAuthentication:iso8601} \tTimestampUnAuthentication:iso8601=%s{TimestampUnAuthentication:iso8601} \tCPUUtilization=%d{CPUUtilization}\tMemUtilization=%d{MemUtilization} \tServiceCount=%d{ServiceCount}\tInterfaceDefRoute=%s{InterfaceDefRoute} \tDefRouteGW=%s{DefRouteGW}\tPrimaryDNSResolver=%s{PrimaryDNSResolver} \tHostStartTime=%s{HostStartTime}\tConnectorStartTime=%s{ConnectorStartTime} \tNumOfInterfaces=%d{NumOfInterfaces}\tBytesRxInterface=%d{BytesRxInterface} \tPacketsRxInterface=%d{PacketsRxInterface}\tErrorsRxInterface=%d{ErrorsRxInterface} \tDiscardsRxInterface=%d{DiscardsRxInterface}\tBytesTxInterface=%d{BytesTxInterface} \tPacketsTxInterface=%d{PacketsTxInterface}\tErrorsTxInterface=%d{ErrorsTxInterface} \tDiscardsTxInterface=%d{DiscardsTxInterface}\tTotalBytesRx=%d{TotalBytesRx} \tTotalBytesTx=%d{TotalBytesTx}\n
Audit logs	LEEF:1.0|Zscaler|ZPA|4.1|%s{auditOperationType}|cat=ZPA_Audit_Log\t devTime=%s{modifiedTime:epoch}\t creationTime=%s{creationTime:iso8601}\t requestId=%s{requestId}\t sessionId=%s{sessionId}\t auditOldValue=%s{auditOldValue}\t auditNewValue=%s{auditNewValue}\t auditOperationType=%s{auditOperationType}\t objectType=%s{objectType}\t objectName=%s{objectName}\t objectId=%d{objectId}\t accountName=%d{customerId}\t usrName=%s{modifiedByUser}\n

- (Optional) You can define a streaming Policy for the log receiver. This entails configuring the SAML Attributes, Application Segments, Segment Groups, Client Types, and Session Statuses. For more information on configuring these settings, see the Log Stream instructions.

f. Click Next.

g. In the Review tab, verify your log receiver configuration.

h. Click Save.

1.6.3 | Ingest authentication logs and data

Abstract

Ingest authentication logs from external authentication services, such as Okta and Azure AD, into authentication stories with Cortex XSIAM.

When you ingest authentication logs and data from an external source, Cortex XSIAM can weave that information into authentication stories. An authentication story unites logs and data regardless of the information source (for example, from an on-premise KDC or from a cloud-based authentication service) into a uniform schema. To search authentication stories, you can use the Query Builder or XQL Search.

Cortex XSIAM can ingest authentication logs and data from various authentication services.

1.6.3.1 | Ingest audit logs from AWS Cloud Trail

Abstract

Take advantage of Cortex XSIAM investigation capabilities and set up audit log ingestion for your AWS CloudTrail logs.

You can forward audit logs for the relative service to Cortex XSIAM from AWS CloudTrail.

To receive audit logs from Amazon Simple Storage Service (Amazon S3) via AWS CloudTrail, you must first configure data collection from Amazon S3. You can then configure the Data Sources settings in Cortex XSIAM for Amazon S3. After you set up collection integration, Cortex XSIAM begins receiving new logs and data from the source.

We do not recommend ingestion of data from an AWS commercial environment into a FedRAMP-certified Cortex XSIAM tenant. However, if you must do so, contact Customer Support for assistance.

> **NOTE:**
>
> For more information on configuring data collection from Amazon S3 using AWS CloudTrail, see the AWS CloudTrail Documentation.

When Cortex XSIAM begins receiving logs, the app automatically creates an Amazon S3 Cortex Query Language (XQL) dataset (`aws_s3_raw`). This enables you to search the logs with XQL Search using the dataset. For example queries, refer to the in-app XQL Library.

For enhanced cloud protection, you can also configure Cortex XSIAM to stitch Amazon S3 audit logs with other Cortex XSIAM authentication stories across all cloud providers using the same format, which you can query with XQL Search using the `cloud_audit_logs` dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules), when relevant, from Amazon S3 logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides the following:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

Prerequisite Steps

Be sure you do the following tasks before you begin configuring data collection from Amazon S3 via AWS CloudTrail.

- Ensure that you have the proper permissions to access AWS CloudTrail and have the necessary permissions to create audit logs. The following permissions in AWS are the minimum requirements for an Amazon S3 bucket and Amazon Simple Queue Service (SQS).
    - Amazon S3 bucket: `GetObject`
    - SQS: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`.
- Determine how you want to provide access to Cortex XSIAM to your logs and to perform API operations. You have the following options:
    - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user. This is the default option as explained in Configure the Amazon S3 collection by selecting Access Key.
    - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. For more information, see Creating a role to delegate permissions to an AWS service. This is the Assumed Role option described in the Amazon S3 collection configuration.

  To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

To configure Cortex XSIAM to receive audit logs from Amazon S3 via AWS Cloudtrail:

1. Log in to the AWS Management Console.
2. From the menu bar, ensure that you have selected the correct region for your configuration.
3. Configure an AWS CloudTrail trail with audit logs.

   > **NOTE:**
   >
   > - For more information on creating an AWS CloudTrail trail, see Create a trail.
   > - If you already have an Amazon S3 bucket configured with AWS CloudTrail audit logs, skip this step and go to Configure an Amazon Simple Queue Service (SQS).

   a. Open the CloudTrail Console, and click Create trail.

   b. Configure the following settings for your CloudTrail trail, where the default settings should be configured unless otherwise indicated.

- Trail name: Specify a descriptive name for your CloudTrail trail.
- Storage location: Select Create new S3 bucket to configure a new Amazon S3 bucket, and specify a unique name in the Trail log bucket and folder field, or select Use existing S3 bucket and Browse to the S3 bucket you already created. If you select an existing Amazon S3 bucket, the bucket policy must grant CloudTrail permission to write to it. For information about manually editing the bucket policy, see Amazon S3 Bucket Policy for CloudTrail.

  > **NOTE:**
  >
  > It is your organization's responsibility to define a retention policy for your Amazon S3 bucket by creating a Lifecycle rule in the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.

- Customer managed AWS KMS key: You can either select a New key and specify the AWS KMS alias, or select an Existing key, and select the AWS KMS alias. The KMS key and S3 bucket must be in the same region.
- SNS notification delivery: (Optional) If you want to be notified whenever CloudTrail publishes a new log to your Amazon S3 bucket, click Enabled. Amazon Simple Notification Service (Amazon SNS) manages these notifications, which are sent for every log file delivery to your S3 bucket, as opposed to every event. When you enable this option, you can either Create a new SNS topic by selecting New and the SNS topic is displayed in the field, or use an Existing topic and select the SNS topic. For more information, see Configure SNS Notifications for CloudTrail.

> **NOTE:**
>
> The CloudWatch Logs - optional settings are not supported and should be left disabled.

c. Click Next, and configure the following Choose log events settings.

- Event type: Leave the default Management events checkbox selected to capture audit logs. Depending on your system requirements, you can also select Data events to log the resource operations performed on or within a resource, or Insights events to identify unusual activity, errors, or user behavior in your account. Based on your selection, additional fields are displayed on the screen to configure under section headings with the same name as the event type.
- Management events section: Configure the following settings.

  -API activity: For Management events, select the API activities you want to log. By default, the Read and Write activities are logged.

  -Exclude AWS KMS events: (Optional) If you want to filter AWS Key Management Service (AWS KMS) events out of your trail, select the checkbox. By default, all AWS KMS events are included.

- Data events section: (Optional) This section is displayed when you configure the Event type to include Data events, which relate to resource operations performed on or within a resource, such as reading and writing to a S3 bucket. For more information on configuring these optional settings in AWS CloudTrail, see Creating a trail.
- Insights events section: (Optional) This section is displayed when you configure the Event type to include Insight events, which relate to unusual activities, errors, or user behavior on your account. For more information on configuring these optional settings in AWS CloudTrail, see Creating a trail.

d. Click Next.

e. In the Review and create page, look over the trail configurations settings that you have configured and if they are correct, click Create trail. If you need to make a change, click Edit beside the particular step that you want to update.

The new trail is listed in the Trails page, which lists the trails in your account from all Regions. It can take up to 15 minutes for CloudTrail to begin publishing log files. You can see the log files in the S3 bucket that you specified. For more information, see Creating a trail.

4. Configure an Amazon Simple Queue Service (SQS).

> **NOTE:**
>
> Ensure that you create your Amazon S3 bucket and Amazon SQS queue in the same region.

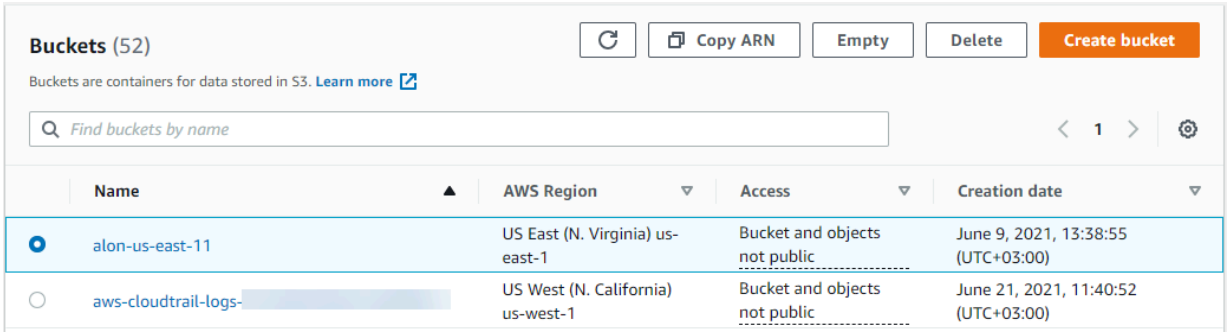
a. In the Amazon SQS Console, click Create Queue.

b. Configure the following settings, where the default settings should be configured unless otherwise indicated.

- Type: Select Standard queue (default).
- Name: Specify a descriptive name for your SQS queue.
- Configuration section: Leave the default settings for the various fields.
- Access policy → Choose method: Select Advanced and update the Access policy code in the editor window to enable your Amazon S3 bucket to publish event notification messages to your SQS queue. Use this sample code as a guide for defining the "`Statement`" with the following definitions:

  -"`Resource`": Leave the automatically generated ARN for the SQS queue that is set in the code, which uses the format "`arn:sns:Region:account-id:topic-name`".

  You can retrieve your bucket's ARN by opening the Amazon S3 Console in a browser window. In the Buckets section, select the bucket that you created for collecting the Amazon S3 flow logs, click Copy ARN, and paste the ARN in the field.



> **NOTE:**
>
> For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    },
  ]
}
```

- Dead-letter queue section: We recommend that you configure a queue for sending undeliverable messages by selecting Enabled, and then in the Choose queue field selecting the queue to send the messages. You may need to create a new queue for this, if you do not already have one set up. For more information, see Amazon SQS dead-letter queues.

c. Click Create queue.

Once the SQS is created, a message indicating that the queue was successfully configured is displayed at the top of the page.

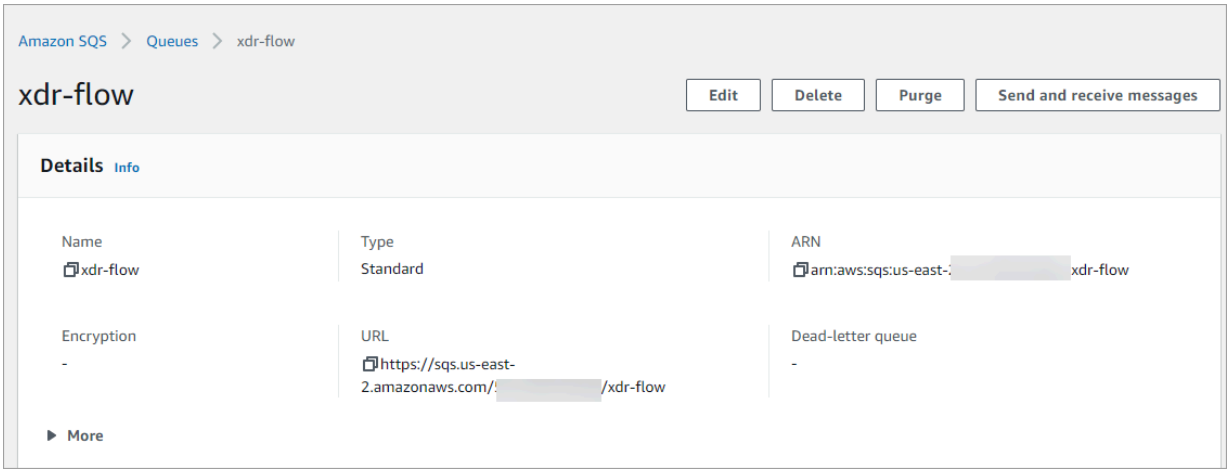
5. Configure an event notification to your Amazon SQS whenever a file is written to your Amazon S3 bucket.

a. Open the Amazon S3 Console and in the Properties tab of your Amazon S3 bucket, scroll down to the Event notifications section, and click Create event notification.

b. Configure the following settings.

- Event name: Specify a descriptive name for your event notification containing up to 255 characters.
- Prefix: Do not set a prefix as the Amazon S3 bucket is meant to be a dedicated bucket for collecting audit logs.
- Event types: Select All object create events for the type of event notifications that you want to receive.
- Destination: Select SQS queue to send notifications to an SQS queue to be read by a server.
- Specify SQS queue: You can either select Choose from your SQS queues and then select the SQS queue, or select Enter SQS queue ARN and specify the ARN in the SQS queue field.

  You can retrieve your SQS queue ARN by opening another instance of the AWS Management Console in a browser window, and opening the Amazon SQS Console, and selecting the Amazon SQS that you created. In the Details section, under ARN, click the copy icon (⧉)), and paste the ARN in the field.



c. Click Save changes.

Once the event notification is created, a message indicating that the event notification was successfully created is displayed at the top of the page.

> **NOTE:**
>
> If your receive an error when trying to save your changes, you should ensure that the permissions are set up correctly.

6. Configure access keys for the AWS IAM user that Cortex XSIAM uses for API operations.

> **NOTE:**
>
> - It your organization's responsibility to ensure that the user who performs this task of creating the access key is designated with the relevant permissions. Otherwise, this can cause the process to fail with errors.
> - Skip this step if you are using an Assumed Role for Cortex XSIAM.

a. Open the AWS IAM Console, and in the navigation pane, select Access management → Users.

b. Select the User name of the AWS IAM user.

c. Select the Security credentials tab, scroll down to the Access keys section, and click Create access key.

d. Click the copy icon next to the Access key ID and Secret access key keys, where you must click Show secret access key to see the secret key and record them somewhere safe before closing the window. You will need to provide these keys when you edit the Access policy of the SQS queue and when setting the AWS Client ID and AWS Client Secret in Cortex XSIAM. If you forget to record the keys and close the window, you will need to generate new keys and repeat this process.

> **NOTE:**
>
> For more information, see Managing access keys for IAM users.

7. Update the Access policy of your Amazon SQS queue.

> **NOTE:**
>
> Skip this step if you are using an Assumed Role for Cortex XSIAM.

a. In the Amazon SQS Console, select the SQS queue that you created in Configure an Amazon Simple Queue Service (SQS).

b. Select the Access policy tab, and Edit the Access policy code in the editor window to enable the IAM user to perform operations on the Amazon SQS with permissions to `SQS:ChangeMessageVisibility`, `SQS:DeleteMessage`, and `SQS:ReceiveMessage`. Use this sample code as a guide for defining the "`Sid": "__receiver_statement`" with the following definitions:

- “aws:SourceArn”: Specify the ARN of the AWS IAM user. You can retrieve the User ARN from the Security credentials tab, which you accessed when configuring access keys for the AWS API user.
- “Resource”: Leave the automatically generated ARN for the SQS queue that is set in the code, which uses the format “arn:sns:Region:account-id:topic-name”.

> **NOTE:**
>
> For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    },
    {
      "Sid": "__receiver_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "[Add the ARN for the AWS IAM user]"
      },
      "Action": [
        "SQS:ChangeMessageVisibility",
        "SQS:DeleteMessage",
        "SQS:ReceiveMessage"
      ],
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]"
    }
  ]
}
```

8. Configure the Amazon S3 collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Amazon S3, and click Connect.

   c. Set these parameters, where the parameters change depending on whether you configured an Access Key or Assumed Role.

   - To provide access to Cortex XSIAM to your logs and perform API operations using a designated AWS IAM user, leave the Access Key option selected. Otherwise, select Assumed Role, and ensure that you Create an Assumed Role for Cortex XSIAM before continuing with these instructions. In addition, when you create an Assumed Role for Cortex XSIAM, ensure that you edit the policy that defines the permissions for the role with the Amazon S3 Bucket ARN and SQS ARN.
   - SQS URL: Specify the SQS URL, which is the ARN of the Amazon SQS that you configured in the AWS Management Console.
   - Name: Specify a descriptive name for your log collection configuration.
   - When setting an Access Key, set these parameters.
     - AWS Client ID: Specify the Access key ID, which you received when you configured access keys for the AWS IAM user in AWS.
     - AWS Client Secret: Specify the Secret access key you received when you configured access keys for the AWS IAM user in AWS.
   - When setting an Assumed Role, set these parameters.
     - Role ARN: Specify the Role ARN for the Assumed Role you created for in AWS.
     - External Id: Specify the External Id for the Assumed Role you created for in AWS.
   - Log Type: Select Audit Logs to configure your log collection to receive audit logs from Amazon S3 via AWS CloudTrail. When configuring audit log collection, the following additional field is displayed for Enhanced Cloud Protection.

     You can Normalize and enrich audit logs by selecting the checkbox. If selected, Cortex XSIAM stitches Amazon S3 audit logs with other Cortex XSIAM authentication stories across all cloud providers using the same format, which you can query with XQL Search using the `cloud_audit_logs` dataset.

   d. Click Test to validate access, and then click Enable.

   Once events start to come in, a green check mark appears underneath the Amazon S3 configuration with the number of logs received.

1.6.3.2 | Ingest logs and data from a GCP Pub/Sub

Abstract

If you use the Pub/Sub messaging service from Global Cloud Platform (GCP), you can send logs and data from GCP to Cortex XSIAM.

If you use the Pub/Sub messaging service from Global Cloud Platform (GCP), you can send logs and data from your GCP instance to Cortex XSIAM. Data from GCP is then searchable in Cortex XSIAM to provide additional information and context to your investigations using the GCP Cortex Query Language (XQL) dataset, which is dependent on the type of GCP logs collected. For example queries, refer to the in-app XQL Library. You can configure a Google Cloud Platform collector to receive generic, flow, audit, or Google Cloud DNS logs. When configuring generic logs, you can receive logs in a Raw, JSON, CEF, LEEF, Cisco, or Corelight format.

You can also configure Cortex XSIAM to normalize different GCP logs as part of the enhanced cloud protection, which you can query with XQL Search using the applicable dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules), when relevant, from GCP logs. While Correlation Rules isssues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only raised on normalized logs.

Enhanced cloud protection provides the following:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

The following table lists the various GCP log types the XQL datasets you can use to query in XQL Search:

GCP Log Type	Dataset	Dataset With Normalized Data
Audit logs, including Google Kubernetes Engine (GKE) audit logs	`google_cloud_logging_raw`	`cloud_audit_logs`
Generic logs	Log Format types: • CEF or `LEEF`: Automatically detected from either the logs or the user's input in the User Interface. • Cisco: `cisco_asa_raw` • Corelight: `corelight_zeek_raw` • JSON or Raw: `google_cloud_logging_raw`	N/A
Google Cloud DNS logs	`google_dns_raw`	`xdr_data`: Once configured, Cortex XSIAM ingests Google Cloud DNS logs as XDR network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`.
Network flow logs	`google_cloud_logging_raw`	`xdr_data`: Once configured, Cortex XSIAM ingests network flow logs as XDR network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`.

NOTE:

When collecting flow logs, we recommend that you include GKE annotations in your logs, which enable you to view the names of the containers that communicated with each other. GKE annotations are only included in logs if appended manually using the custom metadata configuration in GCP. For more information, see VPC Flow Logs Overview. In addition, to customize metadata fields, you must use the gcloud command-line interface or the API. For more information, see Using VPC Flow Logs.

To receive logs and data from GCP, you must first set up log forwarding using a Pub/Sub topic in GCP. You can configure GCP settings using either the GCP web interface or a GCP cloud shell terminal. After you set up your service account in GCP, you configure the Data Collection settings in Cortex XSIAM. The

setup process requires the subscription name and authentication key from your GCP instance.

After you set up log collection, Cortex XSIAM immediately begins receiving new logs and data from GCP.

Set up log forwarding using the GCP web interface

- In Cortex XSIAM, set up Data Collection.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Google Cloud Platform, and click Connect.

    c. Specify the Subscription Name that you previously noted or copied.

    d. Browse to the JSON file containing your authentication key for the service account.

    e. Select the Log Type as one of the following, where your selection changes the options displayed.

- Flow or Audit Logs: When selecting this log type, you can decide whether to normalize and enrich the logs as part of the enhanced cloud protection.
    - (Optional) You can Normalize and enrich flow and audit logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the network flow logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`. In addition, you can configure Cortex XSIAM to normalize GCP audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset.
    - The Vendor is automatically set to Google and Product to Cloud Logging, which is not configurable. This means that all GCP data for the flow and audit logs, whether it's normalized or not, can be queried in XQL Search using the `google_cloud_logging_raw` dataset.
- Generic: When selecting this log type, you can configure the following settings.
    - Log Format: Select the log format type as Raw, JSON, CEF, LEEF, Cisco, or Corelight.
        - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

          > **NOTE:**
          >
          > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the GCP data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the GCP data collector settings. If you did not specify a Vendor or Product in the GCP data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.
        - Cisco: The following fields are automatically set and not configurable.
            - Vendor: Cisco
            - Product: ASA

          Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.
        - Corelight: The following fields are automatically set and not configurable.
            - Vendor: Corelight
            - Product: Zeek

          Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.
        - Raw or JSON: The following fields are automatically set and are configurable.
            - Vendor: Google
            - Product: Cloud Logging

          Raw or JSON data can be queried in XQL Search using the `google_cloud_logging_raw` dataset.

          Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.
    - Vendor: (Optional) Specify a particular vendor name for the GCP generic data collection, which is used in the GCP XQL dataset `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Product: (Optional) Specify a particular product name for the GCP generic data collection, which is used in the GCP XQL dataset name `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Multiline Parsing Regex: (Optional) This option is only displayed when the Log Format is set to Raw, where you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.
- Google Cloud DNS: When selecting this log type, you can configure whether to normalize the logs as part of the enhanced cloud protection.
    - Optional) You can Normalize DNS logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the Google Cloud DNS logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`.
    - The Vendor is automatically set to Google and Product to DNS , which is not configurable. This means that all Google Cloud DNS logs, whether it's normalized or not, can be queried in XQL Search using the `google_dns_raw` dataset.

f. Test the provided settings and, if successful, proceed to Enable log collection.

1. Log in to your GCP account.

2. Set up log forwarding from GCP to Cortex XSIAM.

a. Select Logging → Logs Router.

b. Select Create Sink → Cloud Pub/Sub topic, and then click Next.

c. To filter only specific types of data, select the filter or desired resource.

d. In the Edit Sink configuration, define a descriptive Sink Name.

e. Select Sink Destination → Create new Cloud Pub/Sub topic.

f. Enter a descriptive Name that identifies the sink purpose for Cortex XSIAM, and then Create.

g. Create Sink and then Close when finished.

3. Create a subscription for your Pub/Sub topic.

a. Select the menu icon in G Cloud, and then select Pub/Sub → Topics.

b. Select the name of the topic you created in the previous steps. Use the filters if necessary.

c. Select Create Subscription → Create subscription.

d. Enter a unique Subscription ID.

e. Choose Pull as the Delivery Type.

f. Create the subscription.

After the subscription is set up, G Cloud displays statistics and settings for the service.

g. In the subscription details, identify and note your Subscription Name.

Optionally, use the copy button to copy the name to the clipboard. You will need the name when you configure Collection in Cortex XSIAM.

4. Create a service account and authentication key.

You will use the key to enable Cortex XSIAM to authenticate with the subscription service.

a. Select the menu icon, and then select IAM & Admin → Service Accounts.

b. Create Service Account.

c. Enter a Service account name and then Create.

d. Select a role for the account: Pub/Sub → Pub/Sub Subscriber.

e. Click Continue → Done.

f. Locate the service account by name, using the filters to refine the results, if needed.

g. Click the Actions menu identified by the three dots in the row for the service account and then Create Key.

h. Select JSON as the key type, and then Create.

After you create the service account key, G Cloud automatically downloads it.

5. After Cortex XSIAM begins receiving information from the GCP Pub/Sub service, you can use the XQL Query language to search for specific data.

Set up log forwarding using the GCP cloud shell terminal

- In Cortex XSIAM, set up Data Collection.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Google Cloud Platform, and click Connect.

    c. Specify the Subscription Name that you previously noted or copied.

    d. Browse to the JSON file containing your authentication key for the service account.

    e. Select the Log Type as one of the following, where your selection changes the options displayed.

- Flow or Audit Logs: When selecting this log type, you can decide whether to normalize and enrich the logs as part of the enhanced cloud protection.
    - (Optional) You can Normalize and enrich flow and audit logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the network flow logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`. In addition, you can configure Cortex XSIAM to normalize GCP audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset.
    - The Vendor is automatically set to Google and Product to Cloud Logging, which is not configurable. This means that all GCP data for the flow and audit logs, whether it's normalized or not, can be queried in XQL Search using the `google_cloud_logging_raw` dataset.
- Generic: When selecting this log type, you can configure the following settings.
    - Log Format: Select the log format type as Raw, JSON, CEF, LEEF, Cisco, or Corelight.
        - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

          > **NOTE:**
          >
          > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the GCP data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the GCP data collector settings. If you did not specify a Vendor or Product in the GCP data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.
        - Cisco: The following fields are automatically set and not configurable.
            - Vendor: Cisco
            - Product: ASA

          Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.
        - Corelight: The following fields are automatically set and not configurable.
            - Vendor: Corelight
            - Product: Zeek

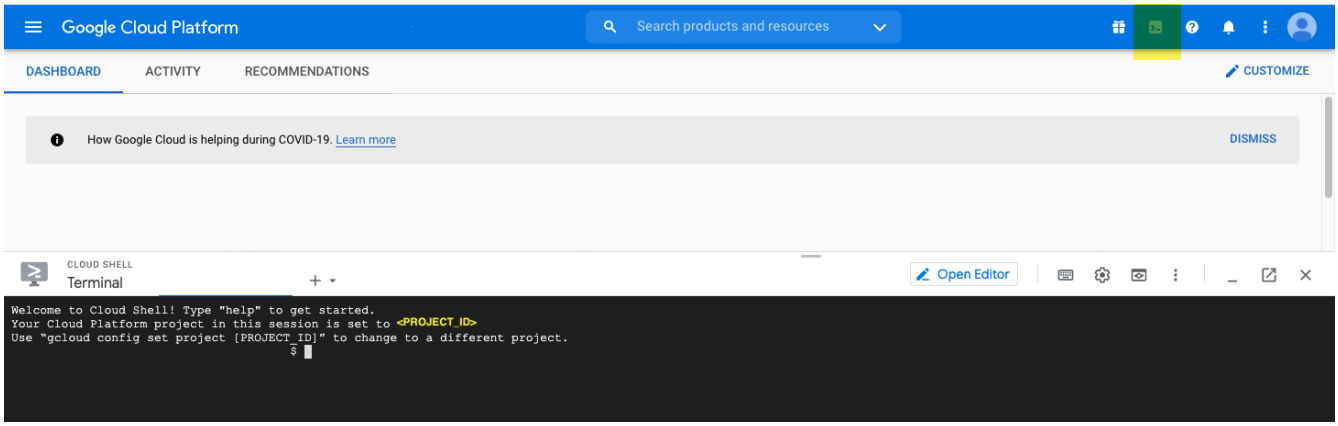
          Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.
        - Raw or JSON: The following fields are automatically set and are configurable.
            - Vendor: Google
            - Product: Cloud Logging

          Raw or JSON data can be queried in XQL Search using the `google_cloud_logging_raw` dataset.

          Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.
    - Vendor: (Optional) Specify a particular vendor name for the GCP generic data collection, which is used in the GCP XQL dataset `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Product: (Optional) Specify a particular product name for the GCP generic data collection, which is used in the GCP XQL dataset name `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Multiline Parsing Regex: (Optional) This option is only displayed when the Log Format is set to Raw, where you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.
- Google Cloud DNS: When selecting this log type, you can configure whether to normalize the logs as part of the enhanced cloud protection.
    - Optional) You can Normalize DNS logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the Google Cloud DNS logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`.
    - The Vendor is automatically set to Google and Product to DNS , which is not configurable. This means that all Google Cloud DNS logs, whether it's normalized or not, can be queried in XQL Search using the `google_dns_raw` dataset.

f. Test the provided settings and, if successful, proceed to Enable log collection.

1. Launch the GCP cloud shell terminal or use your preferred shell with gcloud installed.



2. Define your project ID.

```
gcloud config set project <PROJECT_ID>
```

3. Create a Pub/Sub topic.

```
gcloud pubsub topics create <TOPIC_NAME>
```

4. Create a subscription for this topic.

```
gcloud pubsub subscriptions create <SUBSCRIPTION_NAME> --topic=<TOPIC_NAME>
```

   Note the subscription name you define in this step as you will need it to set up log ingestion from Cortex XSIAM.

5. Create a logging sink.

   During the logging sink creation, you can also define additional log filters to exclude specific logs. To filter logs, supply the optional parameter `--log-filter=<LOG_FILTER>`

```
gcloud logging sinks create <SINK_NAME> pubsub.googleapis.com/projects/<PROJECT_ID>/topics/<TOPIC_NAME> --log-filter=<LOG_FILTER>
```

   If setup is successful, the console displays a summary of your log sink settings:

```
Created [https://logging.googleapis.com/v2/projects/PROJECT_ID/sinks/SINK_NAME]. Please remember to grant `serviceAccount:LOGS_SINK_SERVICE_ACCOUNT` \ the Pub/Sub
Publisher role on the topic. More information about sinks can be found at /logging/docs/export/configure_export
```

6. Grant log sink service account to publish to the new topic.

   Note the `serviceAccount` name from the previous step and use it to define the service for which you want to grant publish access.

```
gcloud pubsub topics add-iam-policy-binding <TOPIC_NAME> --member serviceAccount:<LOGS_SINK_SERVICE_ACCOUNT> --role=roles/pubsub.publisher
```

7. Create a service account.

   For example, use cortex-xdr-sa as the service account name and Cortex XSIAM Service Account as the display name.

```
gcloud iam service-accounts create <SERVICE_ACCOUNT> --description="<DESCRIPTION>" --display-name="<DISPLAY_NAME>"
```

8. Grant the IAM role to the service account.

```
gcloud pubsub subscriptions add-iam-policy-binding <SUBSCRIPTION_NAME> --member serviceAccount:<SERVICE_ACCOUNT>@<PROJECT_ID>.iam.gserviceaccount.com --
role=roles/pubsub.subscriber
```

9. Create a JSON key for the service account.

   You will need the JSON file to enable Cortex XSIAM to authenticate with the GCP service. Specify the file destination and filename using a .json extension.

```
gcloud iam service-accounts keys create <OUTPUT_FILE> --iam-account <SERVICE_ACCOUNT>@<PROJECT_ID>.iam.gserviceaccount.com
```

10. After Cortex XSIAM begins receiving information from the GCP Pub/Sub service, you can use the XQL Query language to search for specific data.

1.6.3.3 | Ingest logs and data from Google Workspace

Abstract

Ingest logs and data from Google Workspace for use in Cortex XSIAM.

Cortex XSIAM can ingest the following types of data from Google Workspace, where most of the data is collected as audit events from various Google reports, using the Google Workspace data collector.

- Google Chrome
- Admin Console
- Google Chat
- Enterprise Groups
- Login
- Rules
- Google drive
- Token
- User Accounts
- SAML
- Alerts
- Emails—Requires a compliance mailbox to ingest email data (not email reports).
    - All message details except email headers and email content (`payload.body`, `payload.parts`, and `snippet`).
    - Attachment details, when Get Attachment Info is selected, includes file name, size, and hash calculation.

The following Google APIs are required to collect the different types of data from Google Workspace.

- For all data types, except emails: Admin SDK API.
- For all data types, except alerts and emails: Admin Reports API (part of Admin SDK API).

    > **NOTE:**
    >
    > For all types of data collected via the Admin Reports API, except alerts and emails, the log events are collected with a preset lag time as reported by Google Workspace. For more information on these lag times for the different types of data, see Google Workspace Data retention and lag times.

- Alerts require implemention of an additional API: Alert Center API (part of Admin SDK API).
- Emails require implemention of the Gmail API.

To receive logs from Google Workspace for any of the data types except emails, you must first enable the Google Workspace Admin SDK API with a user with access to the Admin SDK Reports API. For emails, you must set up a compliance email account as explained in the prerequisite steps below and then enable the Google Workspace Gmail API. Once implemented, you can then configure the Data Sources settings in Cortex XSIAM. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset for the different types of data that you are collecting, which you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library. For all logs, Cortex XSIAM can generate Cortex XSIAM issues for Correlation Rules only, when relevant from Google Workspace logs.

For the different types of data you can collect using the Google Workspace data collector, the following table lists the different datasets, vendors, and products automatically configured, and whether the data is normalized.

Data Type	Dataset	Vendor	Product	Normalized Data
Google Chrome	`google_workspace_chrome_raw`	Google	Workspace Chrome	—
Admin console	`google_workspace_admin_console_raw`	Google	Workspace Admin Console	When relevant, Cortex XSIAM normalizes Admin Console audit logs into authentication stories. All SaaS audit logs are collected in a dataset called `saas_audit_logs` and specific relevant events are collected in the `authentication_story` preset for the `xdr_data` dataset.
Google Chat	`google_workspace_chat_raw`	Google	Workspace Chat	—

Data Type	Dataset	Vendor	Product	Normalized Data
Enterprise groups	google_workspace_enterprise_groups_raw	Google	Workspace Enterprise Groups	When relevant, Cortex XSIAM normalizes Enterprise Group audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
Login	google_workspace_login_raw	Google	Workspace Login	When relevant, Cortex XSIAM normalizes Login audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
Rules	google_workspace_rules_raw	Google	Workspace Rules	When relevant, Cortex XSIAM normalizes Rules audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
Google Drive	google_workspace_drive_raw	Google	Workspace Drive	When relevant, Cortex XSIAM normalizes Google drive audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
Token	google_workspace_token_raw	Google	Workspace Token	When relevant, Cortex XSIAM normalizes Token audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
User accounts	google_workspace_user_accounts_raw	Google	Workspace User Accounts	—
SAML	google_workspace_saml_raw	Google	Workspace SAML	When relevant, Cortex XSIAM normalizes SAML audit logs into authentication stories. All SaaS audit logs are collected in a dataset called saas_audit_logs and specific relevant events are collected in the authentication_story preset for the xdr_data dataset.
Alerts	google_workspace_alerts_raw	Google	Workspace Alerts	—
Emails	google_gmail_raw	Google	Gmail	—

Prerequisite Steps

Be sure you do the following tasks before you begin configuring data collection from Google Workspace using the instructions detailed below.

- When configuring data collection for all data types except emails, you need to complete the Google Workspace Reports API Prerequisites to set up the Google Workspace Admin SDK environment. This entails completing the instructions for Set up the basics and Set up a Google API Console project *without* activating the Reports API service as this will be explained in greater detail in the task below. For more information on these Google Workspace prerequisite steps, see Reports API Prerequisites.
- When you only want to collect Google Workspace alerts without configuring any other data types, you need to set up a Cloud Platform project.
- Before you can collect Google emails, you need to set up the following:

    1. A compliance email account.
    2. The organization's Google Workspace account administrator can now set up a BCC to this compliance email account for all incoming and outgoing emails of any user in the organization.

        a. Login to the Admin direct routing URL in Google Workspace for the user account that you want to configure.

        b. Double-click Routing, and set the following parameters in the Add setting dialog.

            - Routing: Configure the compliance email account that you want to receive a BCC for emails from this user account using the format `BCC TO <compliance email>`. For example, `BCC TO admin@organization.com`.
            - Select Inbound and Outbound to ensure all incoming and outgoing emails are sent.
            - (Optional) To configure another email address to receive a BCC for emails from this account, select Add more recipients in the Also deliver to section, and then click Add.
            - Click Show options, and from the list displayed select Account types to affect → Users.
            - Save your changes.

    This configuration ensures to forward every message sent to a user account to a defined compliance mailbox. After the Google Workspace data collector ingests the emails, they are deleted from the compliance mailbox to prevent email from building up over time (nothing touches the actual users' mailboxes).

    > **NOTE:**
    >
    > - Spam emails from the compliance email account, and from all other monitored email accounts, are not collected.
    > - Any draft emails written in the compliance email account are collected by the Google Workspace data collector, and are then deleted even if the email was never sent.

To set up the Google Workspace integration:

1. Complete the applicable prerequisite steps for the types of data you want to collect from Google Workspace.
2. Log in to your GCP account.
3. Perform Google Workspace Domain-Wide Delegation of Authority when collecting any type of data from Google Workspace except Google Emails.

    When collecting any type of data from Google Workspace except emails, you need to set up Google Workspace enterprise applications to access users' data without any manual authorization. This is performed by following these steps.

    > **NOTE:**
    >
    > For more information on the entire process, see Perform Google Workspace Domain-Wide Delegation of Authority.

    a. Enable the Admin SDK API to create a service account and set credentials for this service account.

    As you complete this step, you need to gather information related to your service account, including the Client ID, Private key file, and Email address, which you will need to use later on in this task.

    1. Select the menu icon → APIs & Services → Library.
    2. Search for the `Admin SDK API`, and select the API from the results list.
    3. Enable the Admin SDK API.
    4. Select APIs & Services → Credentials.
    5. Select + CREATE CREDENTIALS → Service account.
    6. Set the following Service account details in the applicable fields.

- Specify a service account name. This name is automatically used to populate the following field as the service account ID, where the name is changed to lowercase letters and all spaces are changed to hyphens.
- Specify the service account ID, where you can either leave the default service account ID or add a new one. This service account ID is used to set the service account email using the following format: `<id>@<project name>.iam.gserviceaccount.com`.
- (Optional) Specify a service account description.

7. CREATE AND CONTINUE.
8. (Optional) Decide whether you want to Grant this service account access to project or Grant users access to this service account.
9. Click Done.
10. Select your newly created Service Account from the list.
11. Create a service account private key and download the private key file as a JSON file.

    In the Keys tab, select ADD KEY → Create new key, leave the default Key type set to JSON, and CREATE the private key. Once you've downloaded the new private key pair to your machine, ensure that you store it in a secure location, because it's the only copy of this key. You will need to browse to this JSON file when configuring the Google Workplace data collector in Cortex XSIAM.

b. When collecting alerts, enable the Alert Center API to create a service account and set credentials for this service account.

> **NOTE:**
>
> When collecting Google Workspace alerts with other types of data, except emails, you need to configure a service account in Google with the applicable permissions to collect events from the Google Reports API and alerts from the Alert Center API. If you prefer to use different service accounts to collect events and alerts separately, you'll need to create two service accounts with different instances of the Google Workspace data collector. One instance to collect events with a certain service account, and another instance to collect alerts using another service account. The instructions below explain how to set up one Google Workspace instance to collect both event and alerts.

1. Select the menu icon → APIs & Services → Library.
2. Search for the `Alert Center API`, and select the API from the results list.
3. Enable the Alert Center API.
4. Select APIs & Services → Credentials.
5. Select the same service account in the Service Accounts section that you created for the Admin SDK API above.

c. Delegate domain-wide authority to your service account with the Admin Reports API and Alert Center API scopes.

1. Open the Google Admin Console.
2. Select Security → Access and data control → API controls.
3. Scroll down to the Domain wide delegation section, and select MANAGE DOMAIN WIDE DELEGATION.
4. Click Add new.
5. Set the following settings to define permissions for the Admin SDK API.
   - Client ID: Specify the service account's Unique ID, which you can obtain from the Service accounts page by clicking the email of the service account to view further details. When creating a single Google Workspace data collector instance to collect both events and alert data, provide the same service account ID as the Admin SDK API.
   - In the OAuth scopes (comma-delimited) field, paste in the first of the two Admin Reports API scopes: `https://www.googleapis.com/auth/admin.reports.audit.readonly`
   - In the following OAuth scopes (comma-delimited) field, paste in the second Admin Reports API scope: `https://www.googleapis.com/auth/admin.reports.usage.readonly`

     > **NOTE:**
     >
     > For more information on the Admin Reports API scopes, see OAuth 2.0 Scopes for Google APIs.

   - When collecting alerts, add the following Alert Center API scope: `https://www.googleapis.com/auth/apps.alerts`
6. Authorize the domain-wide authority to your service account.

   This ensures that your service account now has domain-wide access to the Google Admin SDK Reports API and Google Workspace Alert Center API, if configured, for all of the users of your domain.

4. Enable the Gmail API to collect Google emails.

When you are configuring the Google Workspace data collector to collect Google emails, the instruction differ depending on whether you are configuring the collection along with other types of data with the Admin SDK API already set up or you are configuring the collection to only include emails using only the Gmail API. The steps below explain both scenarios.

a. Select the menu icon → APIs & Services → Library.

b. Search for the `Gmail API`, and select the API from the results list.

c. Enable the Gmail API.

d. Select APIs & Services → Credentials.

The instructions for setting up credentials differ depending on whether you are setting up the Gmail API together with the Admin SDK API as you are collecting other data types, or you are configuring collection for emails only with the Gmail API.

- When you've already set up the Admin SDK API, verify that the same Service Account that you configured for the Admin SDK API is listed, and continue on to the next step.
- When you're only collecting Google emails without the Admin SDK API, complete these steps.

    1. Select + CREATE CREDENTIALS → Service account.
    2. Set the following Service account details in the applicable fields.

        -Specify a service account name. This name is automatically used to populate the following field as the service account ID, where the name is changed to lowercase letters and all spaces are changed to hyphens.

        -Specify the service account ID, where you can either leave the default service account ID or add a new one. This service account ID is used to set the service account email using the following format: `<id>@<project name>.iam.gserviceaccount.com`.

        -(Optional) Specify a service account description.
    3. CREATE AND CONTINUE.
    4. (Optional) Decide whether you want to Grant this service account access to project or Grant users access to this service account.
    5. Click Done.
    6. Select your newly created Service Account from the list.
    7. Create a service account private key and download the private key file as a JSON file.

        In the Keys tab, select ADD KEY → Create new key, leave the default Key type set to JSON, and CREATE the private key. Once you've downloaded the new private key pair to your machine, ensure that you store it in a secure location as it's the only copy of this key. You will need to browse to this JSON file when configuring the Google Workplace data collector in Cortex XSIAM .

e. Delegate domain-wide authority to your service account with the Gmail API scopes.

1. Open the Google Admin Console.
2. Select Security → Access and data control → API controls.
3. Scroll down to the Domain wide delegation section, and select MANAGE DOMAIN WIDE DELEGATION.

    This step explains how the following Gmail API scopes are added.

- `https://mail.google.com/`
- `https://www.googleapis.com/auth/gmail.addons.current.action.compose`
- `https://www.googleapis.com/auth/gmail.addons.current.message.action`
- `https://www.googleapis.com/auth/gmail.addons.current.message.metadata`
- `https://www.googleapis.com/auth/gmail.addons.current.message.readonly`
- `https://www.googleapis.com/auth/gmail.compose`
- `https://www.googleapis.com/auth/gmail.insert`
- `https://www.googleapis.com/auth/gmail.labels`
- `https://www.googleapis.com/auth/gmail.metadata`
- `https://www.googleapis.com/auth/gmail.modify`
- `https://www.googleapis.com/auth/gmail.readonly`
- `https://www.googleapis.com/auth/gmail.send`
- `https://www.googleapis.com/auth/gmail.settings.basic`
- `https://www.googleapis.com/auth/gmail.settings.sharing`

> **NOTE:**
>
> For more information on the Gmail API scopes, see OAuth 2.0 Scopes for Google APIs.

The instructions differ depending on whether you are setting up the Gmail API together with the Admin SDK API as you are collecting other data types, or you are configuring collection for emails only with the Gmail API.

- When you've already set up the Admin SDK API, Edit the same Service Account that you configured for the Admin SDK API, and add the Gmail API scopes listed above.
- When you're only collecting Google emails without the Admin SDK API, click Add New, and set the following settings to define permissions for the Admin SDK API.

  -Client ID—Specify the service account's Unique ID, which you can obtain from the Service accounts page by clicking the email of the service account to view further details.

  In the OAuth scopes (comma-delimited) field, paste in the first of the Gmail API scopes listed above, and continue adding in the rest of the scopes.

  Authorize the domain-wide authority to your service account.

  This ensures that your service account now has domain-wide access to the Google Gmail API for all of the users of your domain.

5. Prepare your service account to impersonate a user with access to the Admin SDK Reports API when collecting any type of data from Google Workspace except Google emails.

   Only users with access to the Admin APIs can access the Admin SDK Reports API. Therefore, your service account needs to be set up to impersonate one of these users to access the Admin SDK Reports API. This means that when collecting any type of data from Google Workspace except Google emails, you need to designate a user whose Roles permissions are set to access reports, where Security → Reports is selected. This user's email will be required when configuring the Google Workspace data collector in Cortex XSIAM.

   a. In the Google Admin Console, select Directory → Users.

   b. From the list of users listed, select the user configured with the necessary permissions in Admin roles and privileges to view reports, such as a Super Admin, that you want to set up your service account to impersonate.

   c. Record the email of this user as you will need it in Cortex XSIAM .

6. In Cortex XSIAM, select Settings → Data Sources.

7. On the Data Sources page, click Add Data Source, search for and select Google Workspace, and click Connect.

8. Integrate the applicable Google Workspace service with Cortex XSIAM.

   a. Specify a descriptive Name for your log collection integration.

   b. Browse to the JSON file containing your service account key Credentials for the Google Workspace Admin SDK API that you enabled. If you're only collecting Google emails, ensure that you Browse to the JSON file containing your service account private key Credentials for the Gmail API that you enabled.

c. Select the types of data that you want to Collect from Google Workspace.

- Google Chrome: Chrome browser and Chrome OS events included in the Chrome activity reports.
- Admin Console: Account information about different types of administrator activity events included in the Admin console application's activity reports.
- Google Chat: Chat activity events included in the Chat activity reports.
- Enterprise Groups: Enterprise group activity events included in the Enterprise Groups activity reports.
- Login: Account information about different types of login activity events included in the Login application's activity reports.
- Rules: Rules activity events included in the Rules activity report.
- Google drive: Google Drive activity events included in the Google Drive application's activity reports.
- Token: Token activity events included in the Token application's activity reports.
- User Accounts: Account information about different types of User Accounts activity events included in the User Accounts application's activity reports.
- SAML: SAML activity events included in the SAML activity report.
- Alerts: Alerts from the Alert Center API beta version, which is still subject to change.
- Emails: Collects email data (not emails reports). All message details except email headers and email content (`payload.body`, `payload.parts`, and `snippet`).

> **NOTE:**
>
> For more information about the events collected from the various Google Reports, see Google Workspace Reports API Documentation.

For all options selected, except Emails, you must specify the Service Account Email. This is the email account of the user with access to the Admin SDK Reports API that you prepared your service account to impersonate.

When selecting Emails, configure the following.

- Audit Email Account: Specify the email address for the compliance mailbox that you set up.
- Get Attachment Info from the ingested email, which includes file name, size, and hash calculation.

d. Test the connection settings.

To test the connection, you must select one or more log types. Cortex XSIAM then tests the connection settings for the selected log types.

e. If successful, Enable Google Workspace log collection.

1.6.3.4 | Ingest logs from Microsoft Azure Event Hub

Abstract

Ingest logs from Microsoft Azure Event Hub with an option to ingest audit logs to use in Cortex XSIAM authentication stories.

Cortex XSIAM can ingest different types of data from Microsoft Azure Event Hub using the Microsoft Azure Event Hub data collector. To receive logs from Azure Event Hub, you must configure the Data Sources settings in Cortex XSIAM based on your Microsoft Azure Event Hub configuration. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset (`MSFT_Azure_raw`) that you can use to initiate XQL Search queries. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to normalize Azure Event Hub audit logs, including Azure Kubernetes Service (AKS) audit logs, with other Cortex XSIAM authentication stories across all cloud providers using the same format, which you can query with XQL Search using the `cloud_audit_logs` dataset. For logs that you do not configure Cortex XSIAM to normalize, you can change the default dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from Azure Event Hub logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only raised on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

WARNING:

- Misconfiguration of Event Hub resources could cause ingestion delays.
- In an existing Event Hub integration, do not change the mapping to a different Event Hub.
- Do not use the same Event Hub for more than two purposes.

The following table provides a brief description of the different types of Azure audit logs you can collect.

NOTE:

For more information on Azure Event Hub audit logs, see Overview of Azure platform logs.

Type Of Data	Description
Activity logs	Retrieves events related to the operations on each Azure resource in the subscription from the outside in addition to updates on Service Health events. **NOTE:** These logs are from the management plane.
Azure Active Directory (AD) Activity logs and Azure Sign-in logs	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant. **NOTE:** Even though you can collect Azure AD Activity logs and Azure Sign-in logs using the Azure Event Hub data collector, we recommend using the Microsoft Office 365 data collector, because it is easier to configure. In addition, ensure that you do not configure both collectors to collect the same types of logs, because if you do so, you will be creating duplicate data in Cortex XSIAM.
Resource logs, including AKS audit logs	Retrieves events related to operations that were performed within an Azure resource. **NOTE:** These logs are from the data plane.

NOTE:

If you want to ingest raw Microsoft Defender for Endpoint events, use the Microsoft Defender log collector. For more information, see Ingest raw EDR events from Microsoft Defender for Endpoint.

PREREQUISITE:

Ensure that you do the following tasks before you begin configuring data collection from Azure Event Hub.

- Before you set up an Azure Event Hub, calculate the quantity of data that you expect to send to Cortex XSIAM, taking into account potential data spikes and potential increases in data ingestion, because partitions cannot be modified after creation. Use this information to ascertain the optimal number of partitions and Throughput Units (for Azure Basic or Standard) or Processing Units (for Azure Premium). Configure your Event Hub accordingly.
- Create an Azure Event Hub. We recommend using a dedicated Azure Event Hub for this Cortex XSIAM integration. For more information, see Quickstart: Create an event hub using Azure portal.
- Each partition can support a throughput of up to 1 MB/s.
- Ensure the format for the logs you want collected from the Azure Event Hub is either JSON or raw.

Configure the Azure Event Hub collection in Cortex XSIAM:

1. In the Microsoft Azure console, open the Event Hubs page, and select the Azure Event Hub that you created for collection in Cortex XSIAM.
2. Record the following parameters from your configured event hub, which you will need when configuring data collection in Cortex XSIAM.

- Your event hub's consumer group.
    1. Select Entities → Event Hubs, and select your event hub.
    2. Select Entities → Consumer groups, and select your event hub.
    3. In the Consumer group table, copy the applicable value listed in the Name column for your Cortex XSIAM data collection configuration.
- Your event hub's connection string for the designated policy.
    1. Select Settings → Shared access policies.
    2. In the Shared access policies table, select the applicable policy.
    3. Copy the Connection string-primary key.
- Your storage account connection string required for partitions lease management and checkpointing in Cortex XSIAM.
    1. Open the Storage accounts page, and either create a new storage account or select an existing one, which will contain the storage account connection string.
    2. Select Security + networking → Access keys, and click Show keys.
    3. Copy the applicable Connection string.

3. Configure diagnostic settings for the relevant log types you want to collect and then direct these diagnostic settings to the designated Azure Event Hub.

    a. Open the Microsoft Azure console.

    b. Your navigation is dependent on the type of logs you want to configure.

Log Type	**Navigation Path**
Activity logs	Select Azure services → Activity log → Export Activity Logs, and +Add diagnostic setting.
Azure AD Activity logs and Azure Sign-in logs	1. Select Azure services → Azure Active Directory. 2. Select Monitoring → Diagnostic settings, and +Add diagnostic setting.
Resource logs, including AKS audit logs	1. Search for Monitor, and select Settings → Diagnostic settings. 2. From your list of available resources, select the resource that you want to configure for log collection, and then select +Add diagnostic setting. **NOTE:** For every resource that you want to confiure, you'll have to repeat this step, or use Azure policy for a general configuration.

    c. Set the following parameters:

- Diagnostic setting name: Specify a name for your Diagnostic setting.
- Logs Categories/Metrics: The options listed are dependent on the type of logs you want to configure. For Activity logs and Azure AD logs and Azure Sign-in logs, the option is called Logs Categories, and for Resource logs it's called Metrics.

Log Type	Log Categories/Metrics
Activity logs	Select from the list of applicable Activity log categories, the ones that you want to configure your designated resource to collect. We recommend selecting all of the options. ◦ Administrative ◦ Security ◦ ServiceHealth ◦ Alert ◦ Recommendation ◦ Policy ◦ Autoscale ◦ ResourceHealth
Azure AD Activity logs and Azure Sign-in logs	Select from the list of applicable Azure AD Activity and Azure Sign-in Logs Categories, the ones that you want to configure your designated resource to collect. You can select any of the following categories to collect these types of Azure logs. ◦ Azure AD Activity logs: ▪ AuditLogs ◦ Azure Sign-in logs: ▪ SignInLogs ▪ NonInteractiveUserSignInLogs ▪ ServicePrincipalSignInLogs ▪ ManagedIdentitySignInLogs ▪ ADFSSignInLogs **NOTE:** There are additional log categories displayed. We recommend selecting all the available options.
Resource logs, including AKS audit logs	The list displayed is dependent on the resource that you selected. We recommend selecting all the options available for the resource.

- Destination details: Select Stream to event hub, where additional parameters are displayed that you need to configure. Ensure that you set the following parameters using the same settings for the Azure Event Hub that you created for the collection.
  - Subscription: Select the applicable Subscription for the Azure Event Hub.
  - Event hub namespace: Select the applicable Subscription for the Azure Event Hub.
  - (Optional) Event hub name: Specify the name of your Azure Event Hub.
  - Event hub policy: Select the applicable Event hub policy for your Azure Event Hub.

d. Save your settings.

4. Configure the Azure Event Hub collection in Cortex XSIAM.

a. Select Settings → Data Sources.

b. On the Data Sources page, click Add Data Source, search for and select Azure Event Hub, and click Connect.

c. Set these parameters.

- Name: Specify a descriptive name for your log collection configuration.
- Event Hub Connection String: Specify your event hub's connection string for the designated policy.
- Storage Account Connection String: Specify your storage account's connection string for the designated policy.
- Consumer Group: Specify your event hub's consumer group.
- Log Format: Select the log format for the logs collected from the Azure Event Hub as Raw, JSON, CEF, LEEF, Cisco-asa, or Corelight.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the log format is automatically configured. As a result, the Log Format option is removed and is no longer available to configure (default).

  - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

    > **NOTE:**
    >
    > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the Azure Event Hub data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the Azure Event Hub data collector settings. If you did not specify a Vendor or Product in the Azure Event Hub data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

  - Cisco-asa: The following fields are automatically set and not configurable.
    - Vendor: Cisco
    - Product: ASA

    Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.

  - Corelight: The following fields are automatically set and not configurable.
    - Vendor: Corelight
    - Product: Zeek

    Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.

  - Raw or JSON: The following fields are automatically set and are configurable.
    - Vendor: Msft
    - Product: Azure

    Raw or JSON data can be queried in XQL Search using the `msft_azure_raw` dataset.

- Vendor and Product: Specify the Vendor and Product for the type of logs you are ingesting.

  The Vendor and Product are used to define the name of your Cortex Query Language (XQL) dataset (`<vendor>_<product>_raw`). The Vendor and Product values vary depending on the Log Format selected. To uniquely identify the log source, consider changing the values if the values are configurable.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the Vendor and Product fields are automatically configured, so these fields are removed as available options (default).

- Normalize and enrich audit logs: (Optional) For enhanced cloud protection, you can Normalize and enrich audit logs by selecting the checkbox (default). If selected, Cortex XSIAM normalizes and enriches Azure Event Hub audit logs with other Cortex XSIAM authentication stories across all cloud providers using the same format. You can query this normalized data with XQL Search using the `cloud_audit_logs` dataset.

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears underneath the Azure Event Hub configuration with the amount of data received.

1.6.3.5 | Ingest logs from Microsoft Office 365

Abstract

Ingest logs and data from Microsoft Office 365 Management Activity API and Microsoft Graph API for use in Cortex XSIAM.

> **NOTE:**

- Ingesting Microsoft Entra ID (formerly known as Azure AD) authentication and audit events from Microsoft Graph API requires a Microsoft Azure Premium 1 or Premium 2 license. Alternatively, if the directory type is Azure AD B2C, the sign-in reports are accessible through the API without any additional license requirement.
- To ingest **email** logs and data from Microsoft Office 365, use the dedicated data collector. For more information, see Ingest logs and data from Microsoft 365.

Cortex XSIAM can ingest the following logs and data from Microsoft Office 365 Management Activity API and Microsoft Graph API using the Office 365 data collector. Alerts are collected with a delay of 5 minutes. If your organization requires collection that is closer to real-time collection, we recommend using the Microsoft Azure Event Hub integration instead.

To ingest email logs and data from Microsoft Office 365, use the dedicated data collector. For more information, see Ingest logs and data from Microsoft 365.

- Microsoft Office 365 audit events from Management Activity API, which provides information about various user, administrator, system, and policy actions and events from Office 365, Microsoft Entra ID (formerly known as Azure AD) and MDO activity logs.

  > **NOTE:**
  >
  > When auditing is turned off from the default setting, you need to first turn on auditing for your organization to collect Microsoft Office 365 audit events from the Management Activity API. Log duplication of up to 5% in Microsoft products is considered normal. In some cases, such as login to a portal using MFA, two log entries are recorded by design.

- Microsoft Entra ID (Azure AD) authentication and audit events from Microsoft Graph API.

  When collecting Azure AD Authentication Logs, Cortex XSIAM also collects by default all sign-in event types from a beta version of Microsoft Graph API, which is still subject to change. In addition to classic interactive user sign-ins, selecting this option allows you to collect.

  - Non-interactive user sign-ins.
  - Service principal sign-ins.
  - Managed Identities for Azure resource sign-ins.

  > **NOTE:**
  >
  > To address Azure reporting latency, there is a 10-minute latency period for Cortex XSIAM to receive Azure AD logs.

- Microsoft 365 alerts from Microsoft Graph Security API are available for different products.
  - Microsoft Graph Security API v1: Alerts from the following products are available via the Microsoft Graph Security API v1:
    - Microsoft Defender for Cloud, Azure Active Directory Identity Protection, Microsoft Defender for Cloud Apps, Microsoft Defender for Endpoint, Microsoft Defender for Identity, Microsoft 365, Azure Information Protection, and Azure Sentinel.
  - Microsoft Graph Security API v2: Alerts (alerts_v2) from the following products are available via the Microsoft Graph Security API v2 beta version, which is still subject to change:
    - Microsoft 365 Defender unified alerts API, which serves alerts from Microsoft 365 Defender, Microsoft Defender for Endpoint, Microsoft Defender for Office 365, Microsoft Defender for Identity, Microsoft Defender for Cloud Apps, and Microsoft Purview Data Loss Prevention (including any future new signals integrated into M365D).

    > **NOTE:**
    >
    > You can also implement the corresponding Cortex Data Model (XDM) mappings for these Microsoft Graph Security API v2 alerts using Cortex Marketplace via the Microsoft Graph Security content pack.

  To view alerts from the various products via the Microsoft Graph Security API versions, you need to ensure that you've set up the applicable licenses in Office 365. The table below lists the various licenses required for the different Microsoft Defender products. For more information on other Microsoft product licenses, see the Microsoft documentation.

Product	Standalone License	E3 License	E3 + Security Add-On License	E5 License	E5 Security License	E5 Compliance License
Microsoft Defender for Endpoint Plan 1	✓	✓	✓	—	—	—
Microsoft Defender for Endpoint Plan 2	—	—	✓	✓	✓	—
Microsoft Defender for Identity	—	—	✓	✓	✓	—

Product	Standalone License	E3 License	E3 + Security Add-On License	E5 License	E5 Security License	E5 Compliance License
Microsoft Defender for Office 365 Plan 1	✓	—	—	—	—	—
Microsoft Defender for Office 365 Plan 2	✓	—	✓	✓	✓	—
Microsoft Defender for Cloud Apps	—	—	✓	✓	✓	✓

NOTE:

For more information, see the Office 365 Management Activity API schema.

To receive logs from Microsoft Office 365, you must first configure the Data Sources settings in Cortex XSIAM. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset for the different types of logs and data that you are collecting, which you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library. For all Microsoft Office 365 logs, Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules), when relevant, from Office 365 logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

For the different types of data you can collect using the Office 365 data collector, the following table lists the different datasets, vendors, and products automatically configured, and whether the data is normalized.

Data Type	Dataset	Vendor	Product	Normalized Data
Microsoft Office 365 audit events from Management Activity API				
• Microsoft Entra ID (Azure AD)	`msft_o365_azure_ad_raw`	`msft`	`O365 Azure AD`	—
• Exchange Online	`msft_o365_exchange_online_raw`	`msft`	`O365 Exchange Online`	Cortex XSIAM supports normalizing Exchange Online audit logs into stories, which are collected in a dataset called `saas_audit_logs*`.
• SharePoint Online	`msft_o365_sharepoint_online_raw`	`msft`	`O365 Sharepoint Online`	Cortex XSIAM supports normalizing SharePoint Online audit logs into stories, which are collected in a dataset called `saas_audit_logs*`.
• DLP	`msft_o365_dlp_raw`	`msft`	`O365 DLP`	—
• General	`msft_o365_general_raw`	`msft`	`O365 General`	Cortex XSIAM supports normalizing General audit logs into stories, which are collected in a dataset called `saas_audit_logs*`.

Data Type	Dataset	Vendor	Product	Normalized Data
Microsoft Entra ID (Azure AD) authentication events from Microsoft Graph API	`msft_azure_ad_raw`	`msft`	`Azure AD`	When relevant, Cortex XSIAM normalizes Azure AD authentication logs and Azure AD Sign-in logs to authentication stories.
Microsoft Entra ID (Azure AD) audit events from Microsoft Graph API	`msft_azure_ad_audit_raw`	`msft`	`Azure AD Audit`	When relevant, Cortex XSIAM normalizes Azure AD audit logs to cloud audit logs stories.
Alerts from Microsoft Graph Security API v1 and v2	`msft_graph_security_alerts_raw`	`msft`	`Security Alerts`	—

*Note: For the `saas_audit_logs` dataset, the Vendor is saas and Product is Audit Logs.

NOTE:

In FedRAMP environments, Azure sign-in logs are not supported, due to vendor technical constraints.

To set up the Office 365 integration:

1. From the Microsoft Entra ID console (formerly Azure AD console), create an app for Cortex XSIAM with the applicable API permissions for the logs and data you want to collect as detailed in the following table.

Log Type And Data	API/Permission Name
Microsoft Office 365 audit events from Management Activity API	
-Azure AD	Office 365 Management APIs → ActivityFeed.Read
-Exchange Online	Office 365 Management APIs → ActivityFeed.Read
-Sharepoint Online	Office 365 Management APIs → ActivityFeed.Read
-DLP	Office 365 Management APIs → ActivityFeed.ReadDlp
-General	Office 365 Management APIs → ActivityFeed.Read
Azure AD authentication and audit events from Microsoft Graph API	• Microsoft Graph → AuditLog.Read.All • Microsoft Graph → Directory.Read.All
Alerts from Microsoft Graph Security API v1 and v2	• Microsoft Graph → SecurityAlert.Read.All • Microsoft Graph → SecurityEvents.Read.All

   For more information on Microsoft Azure, see the following instructions in the Microsoft documentation portal.

   - Register an app.
   - Add API permissions with type Application.
   - Create an application secret.

2. In Cortex XSIAM, select Settings → Data Sources.

3. On the Data Sources page, click Add Data Source, search for and select Office 365, and click Connect.

4. Integrate the applicable Microsoft Entra ID (Azure AD) service with Cortex XSIAM.

    a. Specify the Tenant Domain of your Microsoft Entra ID tenant.

    b. Obtain the Application Client ID and Secret for your Microsoft Entra ID (Azure AD) service from the Microsoft Entra ID console, and specify the values in Cortex XSIAM.

    These values enable Cortex XSIAM to authenticate with your Microsoft Entra ID (Azure AD) service.

    c. Select the types of logs that you want to receive from Office 365.

    The following options are available.

    - Office 365 Management Activity API
        - Cloud Environment: select the cloud environment used by your organization:
            - Enterprise: Default option for non-US Government tenants
            - GCC: US Government Compliant Cloud tenants
            - GCC High: US Government Compliant Cloud High tenants
            - DoD: US Department of Defense tenants
        - Azure AD: Includes subset of Azure AD audit events and Azure AD authentication events. There can be significant overlap between these and the Azure AD Authentication Logs originating from Microsoft Graph API.

            > **NOTE:**
            >
            > Use this option when you don't want to grant permissions for Azure AD Authentication and Azure AD Audit.

        - Exchange Online: Includes audit logs on Azure Exchange mailboxes and Exchange admin activities on the Office 365 Exchange.
        - Sharepoint Online: Includes audit events on Sharepoint and OneDrive activities.
        - DLP: Includes Microsoft 365 DLP events for Exchange, Sharepoint, and OneDrive.
        - General: Includes audit logs for various Microsoft 365 applications, such as Power BI and Microsoft Forms.
    - Microsoft Graph API
        - Cloud Environment: select the cloud environment used by your organization:
            - Global Service: Default option for non-US Government tenants
            - Government L4: US Government Layer 4 tenants
            - Government L5 (DOD): US Government Layer 5 tenants
        - Azure AD Authentication Logs and Collect all sign-in event types: Azure AD Sign-in logs includes by default all sign-in event types from a beta version of Microsoft Graph API, which is still subject to change. In addition to classic interactive user sign-ins, selecting the Collect all sign-in event types allows you to collect.

            -Non-interactive user sign-ins.

            -Service principal sign-ins.

            -Managed Identities for Azure resource sign-ins.

        - Azure AD Audit Logs: Azure AD Audit logs includes different categories, such as User Management, Group Management and Application Management.
        - Alerts: When this checkbox is selected, alerts from the following products are collected via the Microsoft Graph Security API v1:
            - Microsoft Defender for Cloud, Azure Active Directory Identity Protection, Microsoft Defender for Cloud Apps, Microsoft Defender for Endpoint, Microsoft Defender for Identity, Microsoft 365, Azure Information Protection, and Azure Sentinel.
            - Use Microsoft Graph API v2: When this checkbox is also selected, alerts (alerts_v2) from the following products are only collected via the Microsoft Graph Security API v2 beta version, which is still subject to change:
                - Microsoft 365 Defender unified alerts API, which serves alerts from Microsoft 365 Defender, Microsoft Defender for Endpoint, Microsoft Defender for Office 365, Microsoft Defender for Identity, Microsoft Defender for Cloud Apps, and Microsoft Purview Data Loss Prevention (including any future new signals integrated into M365D).
        - Emails: Deprecated. Use the dedicated email collector instead. For more information, see Ingest logs and data from Microsoft 365.

    d. Click Test to test the connection settings.

To test the connection, you must select one or more log types. Cortex XSIAM then tests the connection settings for the selected log types.

e. If successful, click Enable to enable Office 365 log collection.

1.6.3.6 | Ingest logs and data from Microsoft 365

Abstract

The Microsoft 365 email collector fetches emails through Microsoft Graph API, using an authorized app. A compliance mailbox is not required.

The Microsoft 365 email collector fetches email metadata through Microsoft Graph API, using an authorized app. A compliance mailbox is not required.

LICENSE TYPE:

Email data: The subject, body, and attachments are visible only with an Email Security module license.

NOTE:

For other logs from Microsoft Office 365, use the Office 365 data collector. For more information, see Ingest logs from Microsoft Office 365.

PREREQUISITE:

- A user account with the Microsoft Azure Account Administrator role is required to set up a new Microsoft 365 email collector.
- The following Microsoft Graph API permissions are required:
  - Mailbox access (read-write)
    - Read and write mail in all mailboxes
    - Read contacts in all mailboxes
    - Read all user mailbox settings
  - User information, groups, and directory data (read-only)
    - Read directory data
    - Read all groups
    - Read all users' full profiles

You can narrow down the scope of ingested mailboxes by:

- Microsoft 365 Group
- Distribution List
- Mail-enabled Security Group
- Mail-enabled Users

Datasets

The Microsoft 365 collector ingests data into the following datasets:

- `msft_o365_emails_raw`
- `msft_o365_users_raw`
- `msft_o365_groups_raw`
- `msft_o365_devices_raw`
- `msft_o365_mailboxes_raw`
- `msft_o365_rules_raw`
- `msft_o365_contacts_raw`
- `msft_o365_protected_emails_raw` - requires the Email Security module license.
- `o365_email_threat_submission_policies` - requires the Email Security module license.

Data Encryption

Cortex XSIAM stores email metadata as plain text, and encrypts emails' subject and body. The email body is saved for 48 hours, and then deleted. Analytical detectors analyze raw and encrypted email data, and when necessary, create issues. When an issue with severity higher than or equal to Medium is created for a malicious email, the raw email, including its subject and body (decrypted), is attached to the issue as an artifact. Therefore, you will not be able to perform threat hunting based on email subject and body. Only email metadata such as date, From, or To, are available for threat hunting purposes.

1. In Settings → Data Sources, click Add Data Source, search for and select Microsoft 365, and click Connect.
2. In the wizard that opens, ensure that you have configured the items listed on the Permissions page, and then click Next.
3. To confirm that you know that API authorization consent is required, click OK.
4. Select the Microsoft account from which you want to collect email data.
5. Click Next.
6. Enter your password for the Microsoft account, and click Sign in.
7. If you are asked to perform authentication using your organization's authentication tools, do so.
8. For the list of of permissions that Cortex Email Security requires, click Accept.
9. On the Scope page, select one of the following:
    - Entire organization: Emails will be collected from all mailboxes in your organization.
    - Specific groups: Enter the email addresses of group names, such as Microsoft 365 Groups, Mail-enabled Security Groups, Distribution Lists, or Mail-enabled Users.
10. Click Next.
11. On the Details page, enter a meaningful instance name, and click Next.
12. On the Summary page, check your configurations, and then click Create.

After data starts to come in, a green check mark appears below the Microsoft 365 configuration, along with the amount of data received.

1.6.3.7 | Ingest logs and data from Okta

Abstract

Ingest authentication logs and data from Okta for use in Cortex XSIAM authentication stories.

To receive logs and data from Okta, you must configure the Data Sources settings in Cortex XSIAM. After you set up data collection, Cortex XSIAM immediately begins receiving new logs and data from the source. The information from Okta is then searchable in XQL Search using the `okta_sso_raw` dataset. In addition, depending on the event type, data is normalized to either `xdr_data` or `saas_audit_logs` datasets.

You can collect all types of events from Okta. When setting up the Okta data collector in Cortex XSIAM, a field called Okta Filter is available to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq “user.session.start”.\n`. For Okta information to be woven into authentication stories, “`user.authentication.sso`” events must be collected.

The Okta API enforces concurrent rate limits. The Okta data collector is built with a mechanism which reduces the amount of requests whenever an error is received from the Okta API indicating that too many requests have already been sent. In addition, to ensure you are properly notified about this, an alert is displayed in the Notification Area and a record is added to the Management Audit Logs.

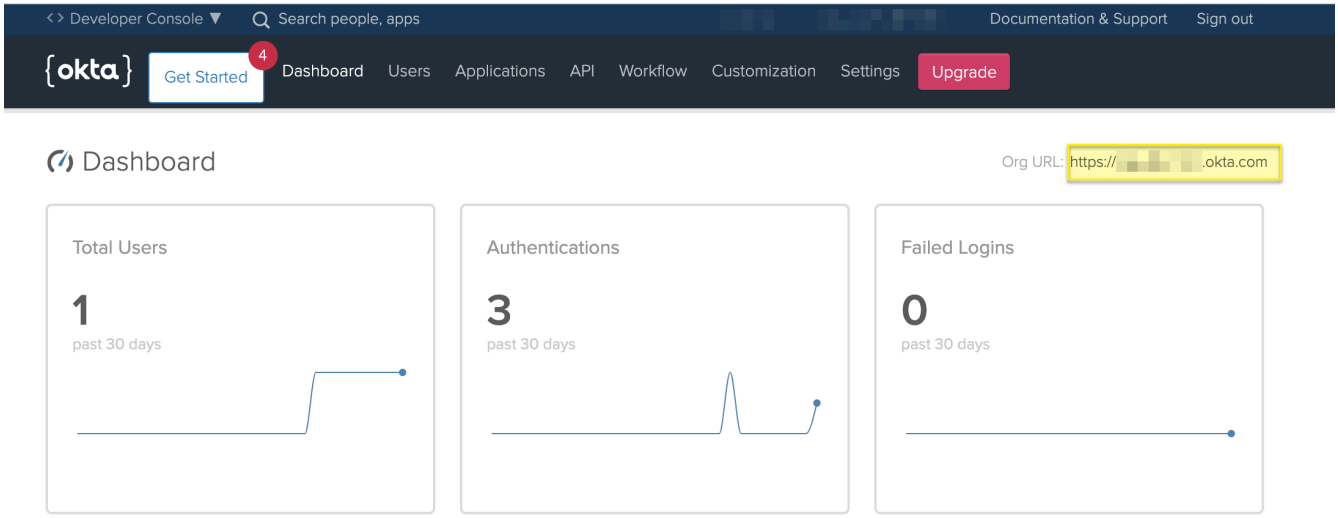
Before you begin configuring data collection from Okta, ensure your Okta user has administrator privileges with a role that can create API tokens, such as the read-only administrator, Super administrator, and Organization administrator. For more information, see the Okta Administrators Documentation.

To configure the Okta collection in Cortex XSIAM:

1. Identify the domain name of your Okta service.

    From the Dashboard of your Okta console, note your Org URL.

    For more information, see the Okta Documentation.



2. Obtain your authentication token in Okta.

    a. Select API → Tokens.

    b. Create Token and record the token value.

    This is your only opportunity to record the value.

3. Select Settings → Data Sources.

4. On the Data Sources page, click Add Data Source, search for and select Okta, and click Connect.

5. Integrate the Okta authentication service with Cortex XSIAM.

    a. Specify the OKTA DOMAIN (Org URL) that you identified on your Okta console.

    b. Specify the TOKEN used to authenticate with Okta.

    c. Specify the Okta Filter to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq “user.session.start”.\n`. For Okta information to be weaved into authentication stories, “`user.authentication.sso`” events must be collected.

    d. Test the connection settings.

    e. If successful, Enable Okta log collection.

    Once events start to come in, a green check mark appears underneath the Okta configuration with the amount of data received.

6. After Cortex XSIAM begins receiving information from the service, you can Create an XQL Query to search for specific data. When including authentication events, you can also Create an Authentication Query to search for specific authentication data.

1.6.3.8 | Ingest logs and data from OneLogin

Abstract

Learn how to ingest different types of logs and data from OneLogin.

Cortex XSIAM can ingest different types of data from OneLogin accounts using the OneLogin data collector.

To receive logs and data from OneLogin via the OneLogin REST APIs, you must configure the Data Sources settings in Cortex XSIAM based on your OneLogin credentials. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset for the different types of data collected and normalizes the ingested data into authentication stories, where specific relevant events are collected in the `authentication_story` preset for the `xdr_data` dataset. You can search these datasets using XQL Search queries. For all logs, Cortex XSIAM can generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC), when relevant from OneLogin logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

The following table provides a description of the different types of data you can collect, the collection method and fetch interval for the data collected, and the name of the dataset to use in Cortex Query Language (XQL) queries.

Data Type	Description	Collection Method	Fetch Interval	Dataset Name
Log collection				
Events	User logins, administrative operations, provisioning, and a list of all OneLogin event types	Appends data	30 seconds	onelogin_events_raw
Directory				
Users	Lists of users	Overwrites data	10 minutes	onelogin_users_raw
Groups	Lists of groups	Overwrites data	10 minutes	onelogin_groups_raw
Apps	Lists of apps	Overwrites data	10 minutes	onelogin_apps_raw

Before you configure Cortex XSIAM data collection from OneLogin, make sure you have the following.

- An Advanced OneLogin account.
- Owner or administrator permissions in your OneLogin account which enable Cortex XSIAM to access the OneLogin account and generate the OAuth 2.0 access token.
- A Cortex XSIAM user account with permissions to Read Log Collections, for example an Instance Administrator.

Configure Cortex XSIAM to receive logs and data from OneLogin.

1. Log in to OneLogin as an account owner or administrator.
2. Under Administration → Developers → API Credentials, Create a New Credential with scope Read All.
3. In the credential details page, copy the Client ID and the Client Secret, and save them somewhere safe. You will need to provide these keys when you configure the OneLogin data collector in Cortex XSIAM .
4. Select Settings → Data Sources.
5. On the Data Sources page, click Add Data Source, search for and select OneLogin, and click Connect.
6. Configure the following parameters.

- Domain: Specify the domain of the OneLogin instance. The domain name must be in the format `https://<subdomain-name>.onelogin.com`.
- Name: Specify a descriptive and unique name for the configuration.
- Client ID: Specify the Client ID for the OneLogin API credential pair.
- Secret: Specify the Client Secret for the OneLogin API credential pair.
- Collect: Select the types of data to collect. By default, all the options are selected.
  - Log Collection
    - Events: Retrieves user logins, administrative operations, provisioning, and OneLogin event types. After normalization, the event types are enriched with the event name and description.

    > **NOTE:**
    >
    > Event data is collected every 30 seconds.

  - Directory
    - Users: Retrieves lists of users.
    - Groups: Retrieves lists of groups.
    - Apps: Retrieves lists of apps.

    > **NOTE:**
    >
    > Inventory data snapshots are collected every 10 minutes.

7. Test the connection settings. If successful, Enable the OneLogin log collection.

   When events start to come in, a green check mark appears underneath the OneLogin configuration.

1.6.3.9 | Ingest authentication logs from PingFederate

Abstract

Ingest authentication logs and data from PingFederate for use in Cortex XSIAM authentication stories.

To receive authentication logs from PingFederate, you must first write Audit and Provisioner Audit Logs to CEF in PingFederate and then set up a Syslog Collector in Cortex XSIAM to receive the logs. After you set up log collection, Cortex XSIAM immediately begins receiving new authentication logs from the source. Cortex XSIAM creates a dataset named `ping_identity_pingfederate_raw`. Logs from PingFederate are searchable in Cortex Query Language (XQL) queries using the dataset and surfaced, when relevant, in authentication stories.

1. Activate the Syslog Collector.
2. Set up PingFederate to write logs in CEF.

   To set up the integration, you must have an account for the PingFederate management dashboard and access to create a subscription for SSO logs.

   In your PingFederate deployment, write audit logs in CEF. During this set up you will need the IP address and port you configured in the Syslog Collector.

3. To search for specific authentication logs or data, you can Create an Authentication Query or use the XQL Search.

1.6.3.10 | Ingest authentication logs and data from PingOne

Abstract

Ingest authentication logs and data from PingOne for Enterprise for use in Cortex XSIAM authentication stories.

To receive authentication logs and data from PingOne for Enterprise, you must first set up a Poll subscription in PingOne and then configure the Collection Integrations settings in Cortex XSIAM. After you set up collection integration, Cortex XSIAM immediately begins receiving new authentication logs and data from the source. These logs and data are then searchable in Cortex XSIAM.

1. Set up PingOne for Enterprise to send logs and data.

   To set up the integration, you must have an account for the PingOne management dashboard and access to create a subscription for SSO logs.

   From the PingOne Dashboard:

   a. Set up a Poll subscription.

1. Select Reporting → Subscriptions → Add Subscription.
2. Enter a NAME for the subscription.
3. Select Poll as the subscription type.
4. Leave the remaining defaults and select Done.

b. Identify your account ID and subscription ID.

1. Select the subscription you just set up and note the part of the poll URL between /reports/ and /poll-subscriptions. This is your PingOne account ID.

   For example:

   `https://admin-api.pingone.com/v3/reports/1234567890asdfghjk-123456-zxcvbn/poll-subscriptions/***-0912348765-4567-98012***/events`

   In this URL, the account ID is `1234567890asdfghjk-123456-zxcvbn`.

2. Next, note the part of the poll URL between /poll-subscriptions/ and /events. This is your subscription ID.

   In the example above, the subscription ID is `***-0912348765-4567-98012***`.

2. Select Settings → Data Sources.
3. On the Data Sources page, click Add Data Source, search for and select PingOne, and click Connect.
4. Connect Cortex XSIAM to your PingOne for Enterprise authentication service.

   a. Enter your PingOne ACCOUNT ID.

   b. Enter your PingOne SUBSCRIPTION ID.

   c. Enter your PingOne USER NAME.

   d. Enter your PingOne PASSWORD.

   e. Test the connection settings.

   f. If successful, Enable PingOne authentication log collection.

   After configuration is complete, Cortex XSIAM begins receiving information from the authentication service. From the Integrations page, you can view the log collection summary.

5. To search for specific authentication logs or data, you can Create an Authentication Query or Create an XQL Query.

1.6.4 | Ingest operation and system logs from cloud providers

Abstract

Learn how to ingest operation and system logs from supported cloud providers into Cortex XSIAM.

You can ingest operation and system logs from supported cloud providers into Cortex XSIAM.

1.6.4.1 | Ingest generic logs from Amazon S3

Abstract

Take advantage of Cortex XSIAM investigation capabilities and set up generic log ingestion for your Amazon S3 logs.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Runtime Security or Data Collection add-on.

You can forward generic logs for the relative service to Cortex XSIAM from Amazon S3.

To receive generic data from Amazon Simple Storage Service (Amazon S3), you must first configure data collection from Amazon S3. You can then configure the Data Sources settings in Cortex XSIAM for Amazon S3. After you set up collection integration, Cortex XSIAM begins receiving new logs and data from the source.

> **NOTE:**
>
> For more information on configuring data collection from Amazon S3, see the Amazon S3 Documentation.

When Cortex XSIAM begins receiving logs, the app automatically creates an Amazon S3 Cortex Query Language (XQL) dataset (`<Vendor>_<Product>_raw`). This enables you to search the logs using XQL Search with the dataset. For example queries, refer to the in-app XQL Library. Cortex XSIAM can also generate

Cortex XSIAM issues (Correlation Rules only), when relevant, from Amazon S3 logs.

> **NOTE:**
>
> You need to set up an Amazon S3 data collector to receive generic logs when collecting logs from BeyondTrust Privilege Management Cloud. For more information, see Ingest logs from BeyondTrust Privilege Management Cloud.

> **NOTE:**
>
> If you want to ingest raw EDR events from SentinelOne DeepVisibility, use the SentinelOne DeepVisibility log collector. For more information, see Ingest raw EDR events from SentinelOne DeepVisibility.

Prerequisites

Perform the following tasks before you begin configuring data collection from Amazon S3:

- Create a dedicated Amazon S3 bucket, which collects the generic logs that you want capture. For more information, see Creating a bucket using the Amazon S3 Console.

  > **NOTE:**
  >
  > It is the customer's responsibility to define a retention policy for your Amazon S3 bucket by creating a Lifecycle rule in the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.

- The logs collected by your dedicated Amazon S3 bucket must adhere to the following guidelines.
  - Each log file must use the 1 log per line format.

    By default, multi-line format is not supported. It can only be used for `raw` format when you specifically configure your environment for that use case.

  - The log format must be compressed as gzip or uncompressed.
  - For best performance, we recommend limiting each file size to up to 50 MB (compressed).
- Ensure that you have at a minimum the following permissions in AWS for an Amazon S3 bucket and Amazon Simple Queue Service (SQS).
  - Amazon S3 bucket: `GetObject`
  - SQS: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`.
- Determine how you want to provide access to Cortex XSIAM to your logs and perform API operations. You have the following options:
  - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user.
  - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. For more information, see Creating a role to delegate permissions to an AWS service. This is the Assumed Role option described in the configure the Amazon S3 collection in Cortex XSIAM. For more information on creating an assumed role for Cortex XSIAM, see Create an assumed role.

  To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

Configure Cortex XSIAM to receive generic logs from Amazon S3:

1. Log in to the AWS Management Console.
2. From the menu bar, ensure that you have selected the correct region for your configuration.
3. Configure an Amazon Simple Queue Service (SQS).

   > **NOTE:**
   >
   > Ensure that you create your Amazon S3 bucket and Amazon SQS queue in the same region.

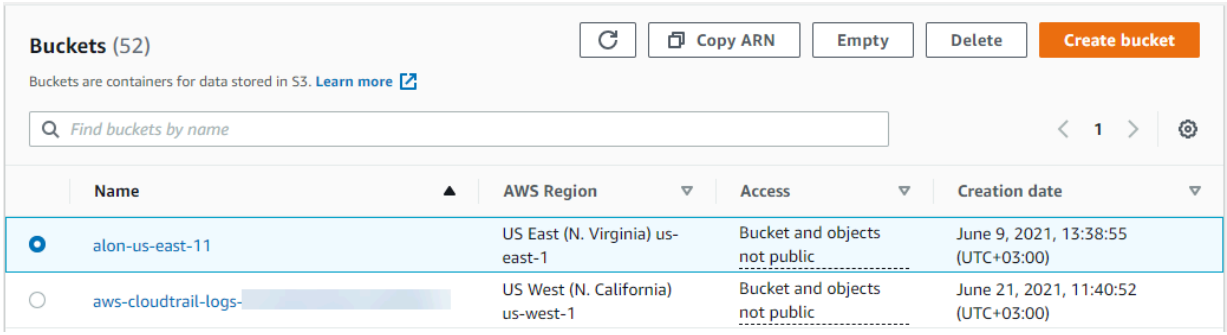
   a. In the Amazon SQS Console, click Create Queue.

   b. Configure the following settings, where the default settings should be configured unless otherwise indicated.

- Type: Select Standard queue (default).
- Name: Specify a descriptive name for your SQS queue.
- Configuration section: Leave the default settings for the various fields.
- Access policy → Choose method: Select Advanced and update the Access policy code in the editor window to enable your Amazon S3 bucket to publish event notification messages to your SQS queue. Use this sample code as a guide for defining the "`Statement`" with the following definitions.

  -"`Resource`": Leave the automatically generated ARN for the SQS queue that is set in the code, which uses the format "`arn:sns:Region:account-id:topic-name`".

  You can retrieve your bucket's ARN by opening the Amazon S3 Console in a browser window. In the Buckets section, select the bucket that you created for collecting the Amazon S3 flow logs, click Copy ARN, and paste the ARN in the field.



> **NOTE:**
>
> For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    }
  ]
}
```

- Dead-letter queue section: We recommend that you configure a queue for sending undeliverable messages by selecting Enabled, and then in the Choose queue field selecting the queue to send the messages. You may need to create a new queue for this, if you do not already have one set up. For more information, see Amazon SQS dead-letter queues.

c. Click Create queue.

Once the SQS is created, a message indicating that the queue was successfully configured is displayed at the top of the page.

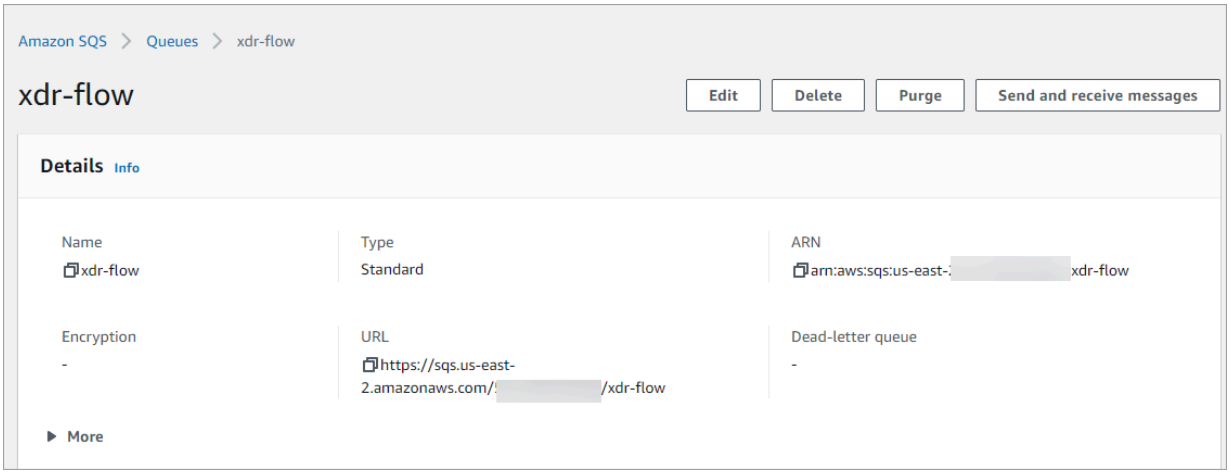
4. Configure an event notification to your Amazon SQS whenever a file is written to your Amazon S3 bucket.

a. Open the Amazon S3 Console and in the Properties tab of your Amazon S3 bucket, scroll down to the Event notifications section, and click Create event notification.

b. Configure the following settings:

- Event name: Specify a descriptive name for your event notification containing up to 255 characters.
- Prefix: Do not set a prefix as the Amazon S3 bucket is meant to be a dedicated bucket for collecting only network flow logs.
- Event types: Select All object create events for the type of event notifications that you want to receive.
- Destination: Select SQS queue to send notifications to an SQS queue to be read by a server.
- Specify SQS queue: You can either select Choose from your SQS queues and then select the SQS queue, or select Enter SQS queue ARN and specify the ARN in the SQS queue field.

  You can retrieve your SQS queue ARN by opening another instance of the AWS Management Console in a browser window, and opening the Amazon SQS Console, and selecting the Amazon SQS that you created. In the Details section, under ARN, click the copy icon ( ), and paste the ARN in the field.



c. Click Save changes.

Once the event notification is created, a message indicating that the event notification was successfully created is displayed at the top of the page.

> **NOTE:**
>
> If your receive an error when trying to save your changes, you should ensure that the permissions are set up correctly.

5. Configure access keys for the AWS IAM user.

> **NOTE:**
>
> - It is the responsibility of your organization to ensure that the user who performs this task of creating the access key is assigned the relevant permissions. Otherwise, this can cause the process to fail with errors.
> - Skip this step if you are using an Assumed Role for Cortex XSIAM.

a. Open the AWS IAM Console, and in the navigation pane, select Access management → Users.

b. Select the User name of the AWS IAM user.

c. Select the Security credentials tab, and scroll down to the Access keys section, and click Create access key.

d. Click the copy icon () next to the Access key ID and Secret access key keys, where you must click Show secret access key to see the secret key, and record them somewhere safe before closing the window. You will need to provide these keys when you edit the Access policy of the SQS queue and when setting the AWS Client ID and AWS Client Secret in Cortex XSIAM. If you forget to record the keys and close the window, you will need to generate new keys and repeat this process.

> **NOTE:**
>
> For more information, see Managing access keys for IAM users.

6. Update the Access policy of your Amazon SQS queue.

> **NOTE:**
>
> Skip this step if you are using an Assumed Role for Cortex XSIAM.

a. In the Amazon SQS Console, select the SQS queue that you created when you configured an Amazon Simple Queue Service (SQS).

b. Select the Access policy tab, and Edit the Access policy code in the editor window to enable the IAM user to perform operations on the Amazon SQS with permissions to `SQS:ChangeMessageVisibility`, `SQS:DeleteMessage`, and `SQS:ReceiveMessage`. Use this sample code as a guide for defining the `“Sid”: “__receiver_statement”` with the following definitions.

- “aws:SourceArn”: Specify the ARN of the AWS IAM user. You can retrieve the User ARN from the Security credentials tab, which you accessed when you configured access keys for the AWS API user.
- “Resource”: Leave the automatically generated ARN for the SQS queue that is set in the code, which uses the format “arn:sns:Region:account-id:topic-name”.

> **NOTE:**
>
> For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    },
    {
      "Sid": "__receiver_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "[Add the ARN for the AWS IAM user]"
      },
      "Action": [
        "SQS:ChangeMessageVisibility",
        "SQS:DeleteMessage",
        "SQS:ReceiveMessage"
      ],
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]"
    }
  ]
}
```

7. Configure the Amazon S3 collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Amazon S3, and click Connect.

   c. Set these parameters, where the parameters change depending on whether you configured an Access Key or Assumed Role.

- To provide access to Cortex XSIAM to your logs and perform API operations using a designated AWS IAM user, leave the Access Key option selected. Otherwise, select Assumed Role, and ensure that you create an Assumed Role for Cortex XSIAM before continuing with these instructions. In addition, when you create an Assumed Role for Cortex XSIAM, ensure that you edit the policy that defines the permissions for the role with the Amazon S3 Bucket ARN and SQS ARN.
- SQS URL: Specify the SQS URL, which is the ARN of the Amazon SQS that you configured in the AWS Management Console.
- Name: Specify a descriptive name for your log collection configuration.
- When setting an Access Key, set these parameters.
    - AWS Client ID: Specify the Access key ID, which you received when you configured access keys for the AWS IAM user in AWS.
    - AWS Client Secret: Specify the Secret access key you received when you configured access keys for the AWS IAM user in AWS.
- When setting an Assumed Role, set these parameters.
    - Role ARN: Specify the Role ARN for the Assumed Role you created for Cortex XSIAM in AWS.
    - External Id: Specify the External Id for the Assumed Role you created for Cortex XSIAM in AWS.
- Log Type: Select Generic to configure your log collection to receive generic logs from Amazon S3, which can include different types of data, such as file and metadata. When selecting this option, the following additional fields are displayed.
    - Log Format: Select the log format type as Raw, JSON, CEF, LEEF, Cisco, Corelight, or Beyondtrust Cloud ECS.

      > **NOTE:**
      >
      > -The Vendor and Product defaults to Auto-Detect when the Log Format is set to CEF or LEEF.
      >
      > -For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the Amazon S3 data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in these fields in the Amazon S3 data collector settings. If you did not specify a Vendor or Product in the Amazon S3 data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

      For a Log Format set to Beyondtrust Cloud ECS, the following fields are automatically set and are not configurable:

      -Vendor: Beyondtrust

      -Product: Privilege Management

      -Compression: Uncompressed

      For more information, see Ingest logs from BeyondTrust Privilege Management Cloud.

      For a Log Format set to Cisco, the following fields are automatically set and not configurable.

      -Vendor: Cisco

      -Product: ASA

      For a Log Format set to Corelight, the following fields are automatically set and not configurable:

      -Vendor: Corelight

      -Product: Zeek

      For a Log Format set to Raw or JSON, the following fields are automatically set and are configurable.

      -Vendor: AMAZON

      -Product: AWS

      Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.
    - Vendor: (Optional) Specify a particular vendor name for the Amazon S3 generic data collection, which is used in the Amazon S3 XQL dataset `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Product: (Optional) Specify a particular product name for the Amazon S3 generic data collection, which is used in the Amazon S3 XQL dataset name `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Compression: Select whether the logs are compressed into a gzip file or are uncompressed.
    - Multiline Parsing Regex: (Optional) This option is only displayed when the Log Format is set to Raw, where you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears underneath the Amazon S3 configuration with the number of logs received.

1.6.4.2 | Ingest logs from Amazon CloudWatch

Abstract

Take advantage of Cortex XSIAM investigation capabilities and set up generic or EKS log ingestion for your Amazon CloudWatch logs.

You can forward generic and Elastic Kubernetes Service (EKS) logs to Cortex XSIAM from Amazon CloudWatch. When forwarding EKS logs, the following log types are included:

- API Server: Logs pertaining to API requests to the cluster.
- Audit: Logs pertaining to cluster access via the Kubernetes API.
- Authenticator: Logs pertaining to authentication requests into the cluster.
- Scheduler: Logs pertaining to scheduling decisions.
- Controller Manager: Logs pertaining to the state of cluster controllers.

You can ingest generic logs of the raw data or in a JSON format from Amazon Kinesis Firehose. EKS logs are automatically ingested in a JSON format from Amazon Kinesis Firehose. To enable log forwarding, you set up Amazon Kinesis Firehose and then add that to your Amazon CloudWatch configuration. After you complete the set up process, logs from the respective service are then searchable in Cortex XSIAM to provide additional information and context to your investigations.

As soon as Cortex XSIAM begins receiving logs, the application automatically creates one of the following Cortex Query Language (XQL) datasets depending on the type of logs you've configured:

- Generic: `<Vendor>_<Product>_raw`
- EKS: `amazon_eks_raw`

These datasets enable you to search the logs in XQL Search. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to normalize EKS audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from AWS logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

Enhanced cloud protection provides the following:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

To set up Amazon CloudWatch integration, you require certain permissions in AWS. You need a role that enables access to configuring Amazon Kinesis Firehose.

1. Set up the Amazon CloudWatch integration in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Amazon CloudWatch, and click Connect.

   c. Specify a descriptive Name for your log collection configuration.

   d. Select the Log Type as one of the following, where your selection changes the options displayed:

- Generic: When selecting this log type, you can configure the following settings:
  - Log Format: Choose the format of the data input source (CloudWatch) that you'll export to Cortex XSIAM , either JSON or Raw.
  - Specify the Vendor and Product for the type of generic logs you are ingesting.

    The vendor and product are used to define the name of your XQL dataset (`<Vendor>_<Product>_raw`). If you do not define a vendor or product, Cortex XSIAM uses the default values of Amazon and AWS with the resulting dataset name as `amazon_aws_raw`. To uniquely identify the log source, consider changing the values.
- EKS: When selecting this log type, the following options are displayed:
  - The Vendor is automatically set to Amazon and Product to EKS , and is non-configurable. This means that all data for the EKS logs, whether it's normalized or not, can be queried in XQL Search using the `amazon_eks_raw` dataset.
  - (Optional) You can decide whether to Normalize and enrich audit logs as part of the enhanced cloud protection by selecting the checkbox (default). If selected, Cortex XSIAM is configured to normalize EKS audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset.

e. Save & Generate Token.

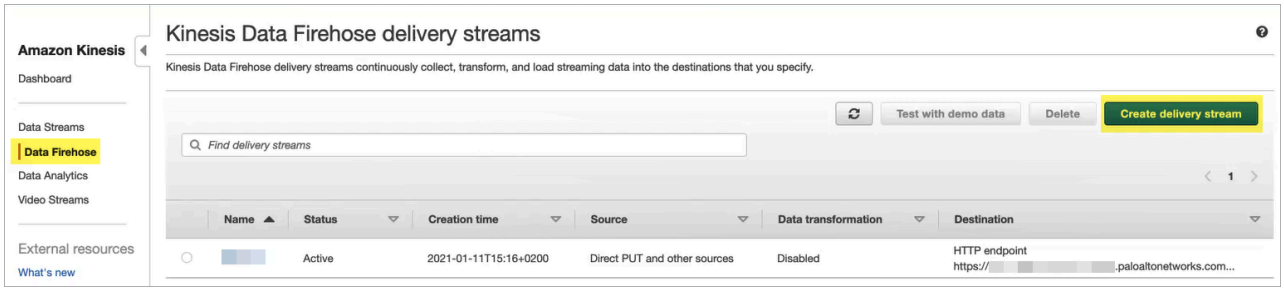
Click the copy icon next to the key and record it somewhere safe. You will need to provide this key when you set up output settings in AWS Kinesis Firehose. If you forget to record the key and close the window you will need to generate a new key and repeat this process.

f. Select Done to close the window.

2. Create a Kinesis Data Firehose delivery stream to your chosen destination.

a. Log in to the AWS Management Console, and open the Kinesis console.

b. Select Data Firehose → Create delivery stream.



c. Define the name and source for your stream.

- Delivery stream name: Enter a descriptive name for your stream configuration.
- Source: Select Direct PUT or other sources.
- Server-side encryption for source records in the delivery stream: Ensure this option is disabled.

Click Next to proceed to the process record configuration.

d. Define the process records.

- Transform source records with AWS Lambda: Set the Data Transformation as Disabled.
- Convert record format: Set Record format conversion as Disabled.

Click Next to proceed to the destination configuration.

e. Choose a destination for the logs.

Choose HTTP Endpoint as the destination and configure the HTTP endpoint configuration settings:

- HTTP endpoint name: Specify the name you used to identify your AWS log collection configuration in Cortex XSIAM.
- HTTP endpoint URL: Copy the API URL associated with your log collection from the Cortex XSIAM management console. The URL will include your tenant name (`https://api-<tenant external URL>/logs/v1/aws`).
- Access key: Paste in the token key you recorded earlier during the configuration of your Cortex XSIAM log collection settings.
- Content encoding: Select GZIP. Disabling content encoding may result in high egress costs.
- Retry duration: Enter 300 seconds.
- S3 bucket: Set the S3 backup mode as Failed data only. For the S3 bucket, we recommend that you create a dedicated bucket for Cortex XSIAM integration.

Click Next to proceed to the settings configuration.

f. Configure additional settings.

- HTTP endpoint buffer conditions: Set the Buffer size as 1 MiB and the Buffer interval as 60 seconds.
- S3 buffer conditions: Use the default settings for Buffer size as 5 MiB and Buffer interval as 300 seconds unless you have alternative sizing preferences.
- S3 compression and encryption: Choose your desired compression and encryption settings.
- Error logging: Select Enabled.
- Permissions: Create or update IAM role option.

Select Next.

g. Review your configuration and Create delivery stream.

When your delivery stream is ready, the status changes from Creating to Active.

3. To begin forwarding logs, add the Kinesis Firehose instance to your Amazon CloudWatch configuration.

   To do this, add a subscription filter for Amazon Kinesis Firehose.

4. Verify the status of the integration.

   Return to the Integrations page and view the statistics for the log collection configuration.

5. After Cortex XSIAM begins receiving logs from your Amazon services, you can use the XQL Search to search for logs in the new dataset.

1.6.4.3 | Ingest logs and data from a GCP Pub/Sub

Abstract

If you use the Pub/Sub messaging service from Global Cloud Platform (GCP), you can send logs and data from GCP to Cortex XSIAM.

If you use the Pub/Sub messaging service from Global Cloud Platform (GCP), you can send logs and data from your GCP instance to Cortex XSIAM. Data from GCP is then searchable in Cortex XSIAM to provide additional information and context to your investigations using the GCP Cortex Query Language (XQL) dataset, which is dependent on the type of GCP logs collected. For example queries, refer to the in-app XQL Library. You can configure a Google Cloud Platform collector to receive generic, flow, audit, or Google Cloud DNS logs. When configuring generic logs, you can receive logs in a Raw, JSON, CEF, LEEF, Cisco, or Corelight format.

You can also configure Cortex XSIAM to normalize different GCP logs as part of the enhanced cloud protection, which you can query with XQL Search using the applicable dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules), when relevant, from GCP logs. While Correlation Rules isssues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only raised on normalized logs.

Enhanced cloud protection provides the following:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

The following table lists the various GCP log types the XQL datasets you can use to query in XQL Search:

GCP Log Type	Dataset	Dataset With Normalized Data
Audit logs, including Google Kubernetes Engine (GKE) audit logs	`google_cloud_logging_raw`	`cloud_audit_logs`

GCP Log Type	Dataset	Dataset With Normalized Data
Generic logs	Log Format types: • CEF or `LEEF`: Automatically detected from either the logs or the user's input in the User Interface. • Cisco: `cisco_asa_raw` • Corelight: `corelight_zeek_raw` • JSON or Raw: `google_cloud_logging_raw`	N/A
Google Cloud DNS logs	`google_dns_raw`	`xdr_data`: Once configured, Cortex XSIAM ingests Google Cloud DNS logs as XDR network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`.
Network flow logs	`google_cloud_logging_raw`	`xdr_data`: Once configured, Cortex XSIAM ingests network flow logs as XDR network connection stories, which you can query with XQL Search using the `xdr_data` dataset with the preset called `network_story`.

NOTE:

When collecting flow logs, we recommend that you include GKE annotations in your logs, which enable you to view the names of the containers that communicated with each other. GKE annotations are only included in logs if appended manually using the custom metadata configuration in GCP. For more information, see VPC Flow Logs Overview. In addition, to customize metadata fields, you must use the gcloud command-line interface or the API. For more information, see Using VPC Flow Logs.

To receive logs and data from GCP, you must first set up log forwarding using a Pub/Sub topic in GCP. You can configure GCP settings using either the GCP web interface or a GCP cloud shell terminal. After you set up your service account in GCP, you configure the Data Collection settings in Cortex XSIAM. The setup process requires the subscription name and authentication key from your GCP instance.

After you set up log collection, Cortex XSIAM immediately begins receiving new logs and data from GCP.

Set up log forwarding using the GCP web interface

- In Cortex XSIAM, set up Data Collection.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Google Cloud Platform, and click Connect.

    c. Specify the Subscription Name that you previously noted or copied.

    d. Browse to the JSON file containing your authentication key for the service account.

    e. Select the Log Type as one of the following, where your selection changes the options displayed.

- Flow or Audit Logs: When selecting this log type, you can decide whether to normalize and enrich the logs as part of the enhanced cloud protection.
    - (Optional) You can Normalize and enrich flow and audit logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the network flow logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`. In addition, you can configure Cortex XSIAM to normalize GCP audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset.
    - The Vendor is automatically set to Google and Product to Cloud Logging, which is not configurable. This means that all GCP data for the flow and audit logs, whether it's normalized or not, can be queried in XQL Search using the `google_cloud_logging_raw` dataset.
- Generic: When selecting this log type, you can configure the following settings.
    - Log Format: Select the log format type as Raw, JSON, CEF, LEEF, Cisco, or Corelight.
        - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

          > **NOTE:**
          >
          > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the GCP data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the GCP data collector settings. If you did not specify a Vendor or Product in the GCP data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

        - Cisco: The following fields are automatically set and not configurable.
            - Vendor: Cisco
            - Product: ASA

          Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.
        - Corelight: The following fields are automatically set and not configurable.
            - Vendor: Corelight
            - Product: Zeek

          Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.
        - Raw or JSON: The following fields are automatically set and are configurable.
            - Vendor: Google
            - Product: Cloud Logging

          Raw or JSON data can be queried in XQL Search using the `google_cloud_logging_raw` dataset.

          Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.
    - Vendor: (Optional) Specify a particular vendor name for the GCP generic data collection, which is used in the GCP XQL dataset `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Product: (Optional) Specify a particular product name for the GCP generic data collection, which is used in the GCP XQL dataset name `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Multiline Parsing Regex: (Optional) This option is only displayed when the Log Format is set to Raw, where you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.
- Google Cloud DNS: When selecting this log type, you can configure whether to normalize the logs as part of the enhanced cloud protection.
    - Optional) You can Normalize DNS logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the Google Cloud DNS logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`.
    - The Vendor is automatically set to Google and Product to DNS , which is not configurable. This means that all Google Cloud DNS logs, whether it's normalized or not, can be queried in XQL Search using the `google_dns_raw` dataset.

f. Test the provided settings and, if successful, proceed to Enable log collection.

1. Log in to your GCP account.

2. Set up log forwarding from GCP to Cortex XSIAM.

a. Select Logging → Logs Router.

b. Select Create Sink → Cloud Pub/Sub topic, and then click Next.

c. To filter only specific types of data, select the filter or desired resource.

d. In the Edit Sink configuration, define a descriptive Sink Name.

e. Select Sink Destination → Create new Cloud Pub/Sub topic.

f. Enter a descriptive Name that identifies the sink purpose for Cortex XSIAM, and then Create.

g. Create Sink and then Close when finished.

3. Create a subscription for your Pub/Sub topic.

a. Select the menu icon in G Cloud, and then select Pub/Sub → Topics.

b. Select the name of the topic you created in the previous steps. Use the filters if necessary.

c. Select Create Subscription → Create subscription.

d. Enter a unique Subscription ID.

e. Choose Pull as the Delivery Type.

f. Create the subscription.

After the subscription is set up, G Cloud displays statistics and settings for the service.

g. In the subscription details, identify and note your Subscription Name.

Optionally, use the copy button to copy the name to the clipboard. You will need the name when you configure Collection in Cortex XSIAM.

4. Create a service account and authentication key.

You will use the key to enable Cortex XSIAM to authenticate with the subscription service.

a. Select the menu icon, and then select IAM & Admin → Service Accounts.

b. Create Service Account.

c. Enter a Service account name and then Create.

d. Select a role for the account: Pub/Sub → Pub/Sub Subscriber.

e. Click Continue → Done.

f. Locate the service account by name, using the filters to refine the results, if needed.

g. Click the Actions menu identified by the three dots in the row for the service account and then Create Key.

h. Select JSON as the key type, and then Create.

After you create the service account key, G Cloud automatically downloads it.

5. After Cortex XSIAM begins receiving information from the GCP Pub/Sub service, you can use the XQL Query language to search for specific data.

Set up log forwarding using the GCP cloud shell terminal

- In Cortex XSIAM, set up Data Collection.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Google Cloud Platform, and click Connect.

    c. Specify the Subscription Name that you previously noted or copied.

    d. Browse to the JSON file containing your authentication key for the service account.

    e. Select the Log Type as one of the following, where your selection changes the options displayed.

- Flow or Audit Logs: When selecting this log type, you can decide whether to normalize and enrich the logs as part of the enhanced cloud protection.
    - (Optional) You can Normalize and enrich flow and audit logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the network flow logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`. In addition, you can configure Cortex XSIAM to normalize GCP audit logs, which you can query with XQL Search using the `cloud_audit_logs` dataset.
    - The Vendor is automatically set to Google and Product to Cloud Logging, which is not configurable. This means that all GCP data for the flow and audit logs, whether it's normalized or not, can be queried in XQL Search using the `google_cloud_logging_raw` dataset.
- Generic: When selecting this log type, you can configure the following settings.
    - Log Format: Select the log format type as Raw, JSON, CEF, LEEF, Cisco, or Corelight.
        - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

          > **NOTE:**
          >
          > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the GCP data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the GCP data collector settings. If you did not specify a Vendor or Product in the GCP data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.
        - Cisco: The following fields are automatically set and not configurable.
            - Vendor: Cisco
            - Product: ASA

          Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.
        - Corelight: The following fields are automatically set and not configurable.
            - Vendor: Corelight
            - Product: Zeek

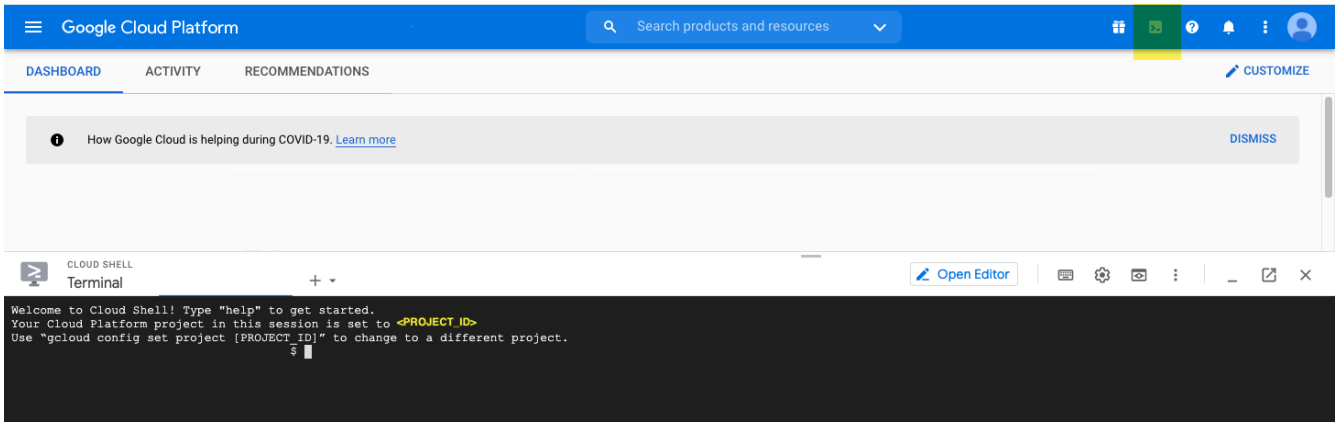
          Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.
        - Raw or JSON: The following fields are automatically set and are configurable.
            - Vendor: Google
            - Product: Cloud Logging

          Raw or JSON data can be queried in XQL Search using the `google_cloud_logging_raw` dataset.

          Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.
    - Vendor: (Optional) Specify a particular vendor name for the GCP generic data collection, which is used in the GCP XQL dataset `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Product: (Optional) Specify a particular product name for the GCP generic data collection, which is used in the GCP XQL dataset name `<Vendor>_<Product>_raw` that Cortex XSIAM creates as soon as it begins receiving logs.
    - Multiline Parsing Regex: (Optional) This option is only displayed when the Log Format is set to Raw, where you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.
- Google Cloud DNS: When selecting this log type, you can configure whether to normalize the logs as part of the enhanced cloud protection.
    - Optional) You can Normalize DNS logs by selecting the checkbox (default). If selected, Cortex XSIAM ingests the Google Cloud DNS logs as Cortex XSIAM network connection stories, which you can query using XQL Search from the `xdr_dataset` dataset with the preset called `network_story`.
    - The Vendor is automatically set to Google and Product to DNS , which is not configurable. This means that all Google Cloud DNS logs, whether it's normalized or not, can be queried in XQL Search using the `google_dns_raw` dataset.

f. Test the provided settings and, if successful, proceed to Enable log collection.

1. Launch the GCP cloud shell terminal or use your preferred shell with gcloud installed.



2. Define your project ID.

```
gcloud config set project <PROJECT_ID>
```

3. Create a Pub/Sub topic.

```
gcloud pubsub topics create <TOPIC_NAME>
```

4. Create a subscription for this topic.

```
gcloud pubsub subscriptions create <SUBSCRIPTION_NAME> --topic=<TOPIC_NAME>
```

   Note the subscription name you define in this step as you will need it to set up log ingestion from Cortex XSIAM.

5. Create a logging sink.

   During the logging sink creation, you can also define additional log filters to exclude specific logs. To filter logs, supply the optional parameter `--log-filter=<LOG_FILTER>`

```
gcloud logging sinks create <SINK_NAME> pubsub.googleapis.com/projects/<PROJECT_ID>/topics/<TOPIC_NAME> --log-filter=<LOG_FILTER>
```

   If setup is successful, the console displays a summary of your log sink settings:

```
Created [https://logging.googleapis.com/v2/projects/PROJECT_ID/sinks/SINK_NAME]. Please remember to grant `serviceAccount:LOGS_SINK_SERVICE_ACCOUNT` \ the Pub/Sub
Publisher role on the topic. More information about sinks can be found at /logging/docs/export/configure_export
```

6. Grant log sink service account to publish to the new topic.

   Note the `serviceAccount` name from the previous step and use it to define the service for which you want to grant publish access.

```
gcloud pubsub topics add-iam-policy-binding <TOPIC_NAME> --member serviceAccount:<LOGS_SINK_SERVICE_ACCOUNT> --role=roles/pubsub.publisher
```

7. Create a service account.

   For example, use cortex-xdr-sa as the service account name and Cortex XSIAM Service Account as the display name.

```
gcloud iam service-accounts create <SERVICE_ACCOUNT> --description="<DESCRIPTION>" --display-name="<DISPLAY_NAME>"
```

8. Grant the IAM role to the service account.

```
gcloud pubsub subscriptions add-iam-policy-binding <SUBSCRIPTION_NAME> --member serviceAccount:<SERVICE_ACCOUNT>@<PROJECT_ID>.iam.gserviceaccount.com --
role=roles/pubsub.subscriber
```

9. Create a JSON key for the service account.

   You will need the JSON file to enable Cortex XSIAM to authenticate with the GCP service. Specify the file destination and filename using a .json extension.

```
gcloud iam service-accounts keys create <OUTPUT_FILE> --iam-account <SERVICE_ACCOUNT>@<PROJECT_ID>.iam.gserviceaccount.com
```

10. After Cortex XSIAM begins receiving information from the GCP Pub/Sub service, you can use the XQL Query language to search for specific data.

1.6.4.4 | Ingest logs from Google Kubernetes Engine

Abstract

Forward your Google Kubernetes Engine (GKE) logs directly to Cortex XSIAM using Elasticsearch Filebeat.

Instead of forwarding Google Kubernetes Engine (GKE) logs directly to Google StackDrive, Cortex XSIAM can ingest container logs from GKE using Elasticsearch Filebeat. To receive logs, you must install Filebeat on your containers and enable Data Collection settings for Filebeat.

When Cortex XSIAM begins receiving logs, the app automatically creates an Cortex Query Language (XQL) dataset using the vendor and product name that you specify during Filebeat setup. It is recommended to specify a descriptive name. For example, if you specify `google` as the vendor and `kubernetes` as the product, the dataset name will be `google_kubernetes_raw`. If you leave the product and vendor blank, Cortex XSIAM assigns the dataset a name of `container_container_raw`.

After Cortex XSIAM creates the dataset, you can search your GKE logs using XQL Search.

1. Install Filebeat on your containers.

   For more information, see https://www.elastic.co/guide/en/beats/filebeat/current/running-on-kubernetes.html.

2. Ingest Logs from Elasticsearch Filebeat.

   Record your token key and API URL for the Filebeat Collector instance as you will need these later in this workflow.

3. Deploy a Filebeat as a DaemonSet on Kubernetes.

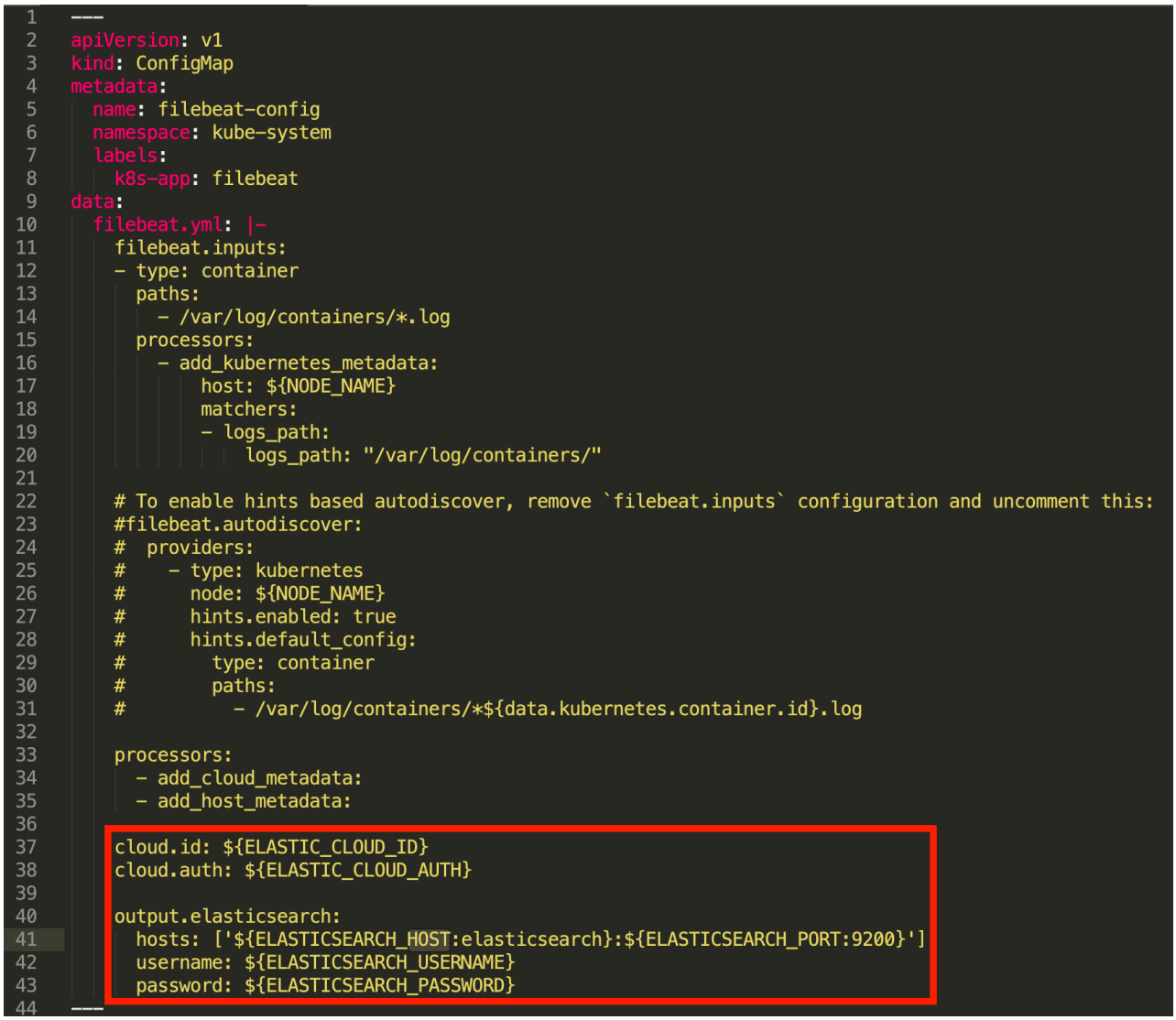
   This ensures there is a running instance of Filebeat on each node of the cluster.

   a. Download the manifest file to a location where you can edit it.

   ```
   curl -L -O https://raw.githubusercontent.com/elastic/beats/7.10/deploy/kubernetes/filebeat-kubernetes.yaml
   ```

   b. Open the YAML file in your preferred text editor.

   c. Remove the `cloud.id` and `cloud.auth` lines.



```yaml
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeat-config
  namespace: kube-system
  labels:
    k8s-app: filebeat
data:
  filebeat.yml: |-
    filebeat.inputs:
    - type: container
      paths:
        - /var/log/containers/*.log
      processors:
        - add_kubernetes_metadata:
            host: ${NODE_NAME}
            matchers:
            - logs_path:
                logs_path: "/var/log/containers/"

    # To enable hints based autodiscover, remove `filebeat.inputs` configuration and uncomment this:
    #filebeat.autodiscover:
    #  providers:
    #    - type: kubernetes
    #      node: ${NODE_NAME}
    #      hints.enabled: true
    #      hints.default_config:
    #        type: container
    #        paths:
    #          - /var/log/containers/*${data.kubernetes.container.id}.log

    processors:
      - add_cloud_metadata:
      - add_host_metadata:

    cloud.id: ${ELASTIC_CLOUD_ID}
    cloud.auth: ${ELASTIC_CLOUD_AUTH}

    output.elasticsearch:
      hosts: ['${ELASTICSEARCH_HOST:elasticsearch}:${ELASTICSEARCH_PORT:9200}']
      username: ${ELASTICSEARCH_USERNAME}
      password: ${ELASTICSEARCH_PASSWORD}
---
```

   d. For the `output.elasticsearch` configuration, replace the `hosts`, `username`, and `password` with environment variable references for `hosts` and `api_key`, and add a field and value for `compression_level` and `bulk_max_size`.

```
output.elasticsearch:
  hosts: ['${ELASTICSEARCH_ENDPOINT}']
  api_key: ${ELASTICSEARCH_API_KEY}
  compression_level: 5
  bulk_max_size: 1000
```

e. In the `DaemonSet` configuration, locate the `env` configuration and replace `ELASTIC_CLOUD_AUTH`, `ELASTIC_CLOUD_ID`, `ELASTICSEARCH_USERNAME`, `ELASTICSEARCH_PASSWORD`, `ELASTICSEARCH_HOST`, `ELASTICSEARCH_PORT` and their relative values with the following.

- `ELASTICSEARCH_ENDPOINT`: Specify the API URL for your Cortex XSIAM tenant. You can copy the URL from the Filebeat Collector instance you set up for GKE in the Cortex XSIAM management console (Settings → (⚙) → Configurations → Data Collection → Custom Collectors → Copy API URL. The URL will include your tenant name (`https://api-tenant external URL:443/logs/v1/filebeat`)
- `ELASTICSEARCH_API_KEY`: Specify the token key you recorded earlier during the configuration of your Filebeat Collector instance.

After you configure these settings your configuration should look like the following image.

```
40  ---
41  apiVersion: apps/v1
42  kind: DaemonSet
43  metadata:
44    name: filebeat
45    namespace: kube-system
46    labels:
47      k8s-app: filebeat
48  spec:
49    selector:
50      matchLabels:
51        k8s-app: filebeat
52    template:
53      metadata:
54        labels:
55          k8s-app: filebeat
56      spec:
57        serviceAccountName: filebeat
58        terminationGracePeriodSeconds: 30
59        hostNetwork: true
60        dnsPolicy: ClusterFirstWithHostNet
61        containers:
62        - name: filebeat
63          image: docker.elastic.co/beats/filebeat:7.10.1
64          args: [
65            "-c", "/etc/filebeat.yml",
66            "-e",
67          ]
68          env:
69          - name: ELASTICSEARCH_ENDPOINT
70            value: <api_url>
71          - name: ELASTICSEARCH_API_KEY
72            value: <api_key>
73          - name: NODE_NAME
74            valueFrom:
75              fieldRef:
76                fieldPath: spec.nodeName
```

f. Save your changes.

4. If you use RedHat OpenShift, you must also specify additional settings.

   See https://www.elastic.co/guide/en/beats/filebeat/7.10/running-on-kubernetes.html.

5. Deploy Filebeat on your Kubernetes.

   `kubectl create -f filebeat-kubernetes.yaml`

   This deploys Filebeat in the kube-system namespace. If you want to deploy the Filebeat configuration in other namespaces, change the namespace values in the YAML file (in any YAML inside this file) and add `-n <your_namespace>`.

After you deploy your configuration, the Filebeat DameonSet runs throughout your containers to forward logs to Cortex XSIAM. You can review the configuration from the Kubernetes Engine console: Workloads → Filebeat → YAML.

> **NOTE:**
>
> Cortex XSIAM supports logs in single line format or multiline format. For more information on handling messages that span multiple lines of text in Elasticsearch Filebeat, see Manage Multiline Messages.

6. After Cortex XSIAM begins receiving logs from GKE, you can use the XQL Search to search for logs in the new dataset.

1.6.4.5 | Ingest logs from Microsoft Azure Event Hub

Abstract

Ingest logs from Microsoft Azure Event Hub with an option to ingest audit logs to use in Cortex XSIAM authentication stories.

Cortex XSIAM can ingest different types of data from Microsoft Azure Event Hub using the Microsoft Azure Event Hub data collector. To receive logs from Azure Event Hub, you must configure the Data Sources settings in Cortex XSIAM based on your Microsoft Azure Event Hub configuration. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset (`MSFT_Azure_raw`) that you can use to initiate XQL Search queries. For example, queries refer to the in-app XQL Library. For enhanced cloud protection, you can also configure Cortex XSIAM to normalize Azure Event Hub audit logs, including Azure Kubernetes Service (AKS) audit logs, with other Cortex XSIAM authentication stories across all cloud providers using the same format, which you can query with XQL Search using the `cloud_audit_logs` dataset. For logs that you do not configure Cortex XSIAM to normalize, you can change the default dataset. Cortex XSIAM can also generate Cortex XSIAM issues (Analytics, IOC, BIOC, and Correlation Rules) when relevant from Azure Event Hub logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only raised on normalized logs.

Enhanced cloud protection provides:

- Normalization of cloud logs
- Cloud logs stitching
- Enrichment with cloud data
- Detection based on cloud analytics
- Cloud-tailored investigations

> **WARNING:**
>
> - Misconfiguration of Event Hub resources could cause ingestion delays.
> - In an existing Event Hub integration, do not change the mapping to a different Event Hub.
> - Do not use the same Event Hub for more than two purposes.

The following table provides a brief description of the different types of Azure audit logs you can collect.

> **NOTE:**
>
> For more information on Azure Event Hub audit logs, see Overview of Azure platform logs.

Type Of Data	Description
Activity logs	Retrieves events related to the operations on each Azure resource in the subscription from the outside in addition to updates on Service Health events. **NOTE:** These logs are from the management plane.
Azure Active Directory (AD) Activity logs and Azure Sign-in logs	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant. **NOTE:** Even though you can collect Azure AD Activity logs and Azure Sign-in logs using the Azure Event Hub data collector, we recommend using the Microsoft Office 365 data collector, because it is easier to configure. In addition, ensure that you do not configure both collectors to collect the same types of logs, because if you do so, you will be creating duplicate data in Cortex XSIAM.

Type Of Data	Description
Resource logs, including AKS audit logs	Retrieves events related to operations that were performed within an Azure resource. **NOTE:** These logs are from the data plane.

NOTE:

If you want to ingest raw Microsoft Defender for Endpoint events, use the Microsoft Defender log collector. For more information, see Ingest raw EDR events from Microsoft Defender for Endpoint.

PREREQUISITE:

Ensure that you do the following tasks before you begin configuring data collection from Azure Event Hub.

- Before you set up an Azure Event Hub, calculate the quantity of data that you expect to send to Cortex XSIAM, taking into account potential data spikes and potential increases in data ingestion, because partitions cannot be modified after creation. Use this information to ascertain the optimal number of partitions and Throughput Units (for Azure Basic or Standard) or Processing Units (for Azure Premium). Configure your Event Hub accordingly.
- Create an Azure Event Hub. We recommend using a dedicated Azure Event Hub for this Cortex XSIAM integration. For more information, see Quickstart: Create an event hub using Azure portal.
- Each partition can support a throughput of up to 1 MB/s.
- Ensure the format for the logs you want collected from the Azure Event Hub is either JSON or raw.

Configure the Azure Event Hub collection in Cortex XSIAM:

1. In the Microsoft Azure console, open the Event Hubs page, and select the Azure Event Hub that you created for collection in Cortex XSIAM.
2. Record the following parameters from your configured event hub, which you will need when configuring data collection in Cortex XSIAM.
   - Your event hub's consumer group.
     1. Select Entities → Event Hubs, and select your event hub.
     2. Select Entities → Consumer groups, and select your event hub.
     3. In the Consumer group table, copy the applicable value listed in the Name column for your Cortex XSIAM data collection configuration.
   - Your event hub's connection string for the designated policy.
     1. Select Settings → Shared access policies.
     2. In the Shared access policies table, select the applicable policy.
     3. Copy the Connection string-primary key.
   - Your storage account connection string required for partitions lease management and checkpointing in Cortex XSIAM.
     1. Open the Storage accounts page, and either create a new storage account or select an existing one, which will contain the storage account connection string.
     2. Select Security + networking → Access keys, and click Show keys.
     3. Copy the applicable Connection string.
3. Configure diagnostic settings for the relevant log types you want to collect and then direct these diagnostic settings to the designated Azure Event Hub.

   a. Open the Microsoft Azure console.

   b. Your navigation is dependent on the type of logs you want to configure.

Log Type	Navigation Path
Activity logs	Select Azure services → Activity log → Export Activity Logs, and +Add diagnostic setting.
Azure AD Activity logs and Azure Sign-in logs	1. Select Azure services → Azure Active Directory. 2. Select Monitoring → Diagnostic settings, and +Add diagnostic setting.

Log Type	Navigation Path
Resource logs, including AKS audit logs	1. Search for Monitor, and select Settings → Diagnostic settings. 2. From your list of available resources, select the resource that you want to configure for log collection, and then select +Add diagnostic setting. **NOTE:** For every resource that you want to confiure, you'll have to repeat this step, or use Azure policy for a general configuration.

c. Set the following parameters:

- Diagnostic setting name: Specify a name for your Diagnostic setting.
- Logs Categories/Metrics: The options listed are dependent on the type of logs you want to configure. For Activity logs and Azure AD logs and Azure Sign-in logs, the option is called Logs Categories, and for Resource logs it's called Metrics.

Log Type	Log Categories/Metrics
Activity logs	Select from the list of applicable Activity log categories, the ones that you want to configure your designated resource to collect. We recommend selecting all of the options. ◦ Administrative ◦ Security ◦ ServiceHealth ◦ Alert ◦ Recommendation ◦ Policy ◦ Autoscale ◦ ResourceHealth
Azure AD Activity logs and Azure Sign-in logs	Select from the list of applicable Azure AD Activity and Azure Sign-in Logs Categories, the ones that you want to configure your designated resource to collect. You can select any of the following categories to collect these types of Azure logs. ◦ Azure AD Activity logs: ▪ AuditLogs ◦ Azure Sign-in logs: ▪ SignInLogs ▪ NonInteractiveUserSignInLogs ▪ ServicePrincipalSignInLogs ▪ ManagedIdentitySignInLogs ▪ ADFSSignInLogs **NOTE:** There are additional log categories displayed. We recommend selecting all the available options.
Resource logs, including AKS audit logs	The list displayed is dependent on the resource that you selected. We recommend selecting all the options available for the resource.

- Destination details: Select Stream to event hub, where additional parameters are displayed that you need to configure. Ensure that you set the following parameters using the same settings for the Azure Event Hub that you created for the collection.
  - Subscription: Select the applicable Subscription for the Azure Event Hub.
  - Event hub namespace: Select the applicable Subscription for the Azure Event Hub.
  - (Optional) Event hub name: Specify the name of your Azure Event Hub.
  - Event hub policy: Select the applicable Event hub policy for your Azure Event Hub.

d. Save your settings.

4. Configure the Azure Event Hub collection in Cortex XSIAM.

a. Select Settings → Data Sources.

b. On the Data Sources page, click Add Data Source, search for and select Azure Event Hub, and click Connect.

c. Set these parameters.

- Name: Specify a descriptive name for your log collection configuration.
- Event Hub Connection String: Specify your event hub's connection string for the designated policy.
- Storage Account Connection String: Specify your storage account's connection string for the designated policy.
- Consumer Group: Specify your event hub's consumer group.
- Log Format: Select the log format for the logs collected from the Azure Event Hub as Raw, JSON, CEF, LEEF, Cisco-asa, or Corelight.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the log format is automatically configured. As a result, the Log Format option is removed and is no longer available to configure (default).

  - CEF or LEEF: The Vendor and Product defaults to Auto-Detect.

    > **NOTE:**
    >
    > For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the Azure Event Hub data collector settings. Yet, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the Azure Event Hub data collector settings. If you did not specify a Vendor or Product in the Azure Event Hub data collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

  - Cisco-asa: The following fields are automatically set and not configurable.
    - Vendor: Cisco
    - Product: ASA

    Cisco data can be queried in XQL Search using the `cisco_asa_raw` dataset.

  - Corelight: The following fields are automatically set and not configurable.
    - Vendor: Corelight
    - Product: Zeek

    Corelight data can be queried in XQL Search using the `corelight_zeek_raw` dataset.

  - Raw or JSON: The following fields are automatically set and are configurable.
    - Vendor: Msft
    - Product: Azure

    Raw or JSON data can be queried in XQL Search using the `msft_azure_raw` dataset.

- Vendor and Product: Specify the Vendor and Product for the type of logs you are ingesting.

  The Vendor and Product are used to define the name of your Cortex Query Language (XQL) dataset (`<vendor>_<product>_raw`). The Vendor and Product values vary depending on the Log Format selected. To uniquely identify the log source, consider changing the values if the values are configurable.

  > **NOTE:**
  >
  > When you Normalize and enrich audit logs, the Vendor and Product fields are automatically configured, so these fields are removed as available options (default).

- Normalize and enrich audit logs: (Optional) For enhanced cloud protection, you can Normalize and enrich audit logs by selecting the checkbox (default). If selected, Cortex XSIAM normalizes and enriches Azure Event Hub audit logs with other Cortex XSIAM authentication stories across all cloud providers using the same format. You can query this normalized data with XQL Search using the `cloud_audit_logs` dataset.

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears underneath the Azure Event Hub configuration with the amount of data received.

1.6.4.6 | Ingest logs and data from Okta

Abstract

Ingest authentication logs and data from Okta for use in Cortex XSIAM authentication stories.

To receive logs and data from Okta, you must configure the Data Sources settings in Cortex XSIAM. After you set up data collection, Cortex XSIAM immediately begins receiving new logs and data from the source. The information from Okta is then searchable in XQL Search using the `okta_sso_raw` dataset. In addition, depending on the event type, data is normalized to either `xdr_data` or `saas_audit_logs` datasets.

You can collect all types of events from Okta. When setting up the Okta data collector in Cortex XSIAM, a field called Okta Filter is available to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq "user.session.start".\n`. For Okta information to be woven into authentication stories, "`user.authentication.sso`" events must be collected.

The Okta API enforces concurrent rate limits. The Okta data collector is built with a mechanism which reduces the amount of requests whenever an error is received from the Okta API indicating that too many requests have already been sent. In addition, to ensure you are properly notified about this, an alert is displayed in the Notification Area and a record is added to the Management Audit Logs.

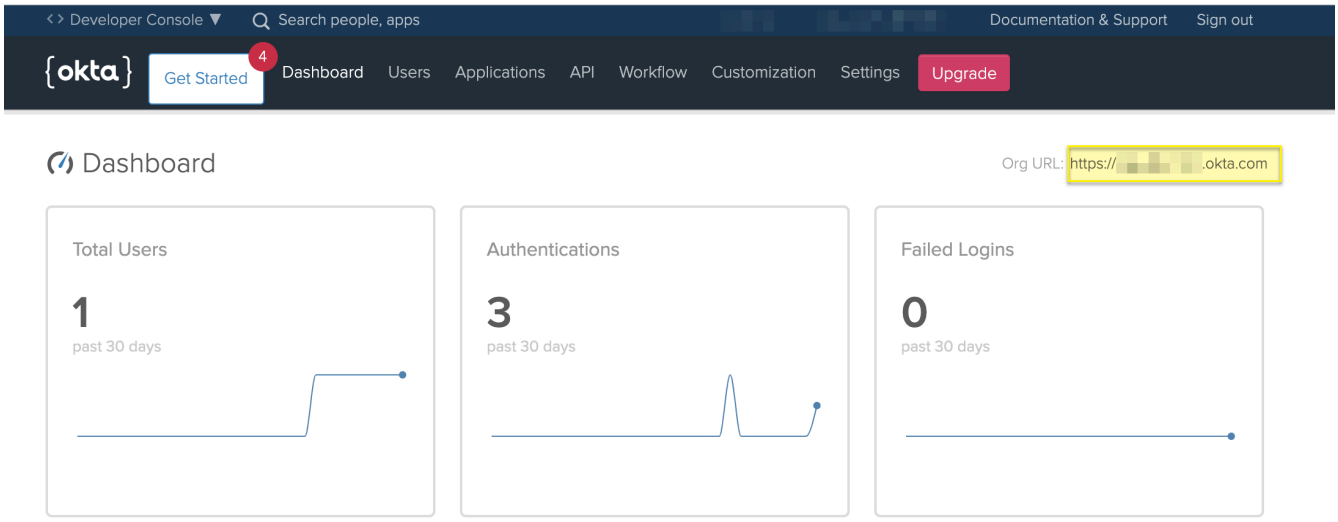
Before you begin configuring data collection from Okta, ensure your Okta user has administrator privileges with a role that can create API tokens, such as the read-only administrator, Super administrator, and Organization administrator. For more information, see the Okta Administrators Documentation.

To configure the Okta collection in Cortex XSIAM:

1. Identify the domain name of your Okta service.

   From the Dashboard of your Okta console, note your Org URL.

   For more information, see the Okta Documentation.



2. Obtain your authentication token in Okta.

   a. Select API → Tokens.

   b. Create Token and record the token value.

      This is your only opportunity to record the value.

3. Select Settings → Data Sources.

4. On the Data Sources page, click Add Data Source, search for and select Okta, and click Connect.

5. Integrate the Okta authentication service with Cortex XSIAM.

   a. Specify the OKTA DOMAIN (Org URL) that you identified on your Okta console.

   b. Specify the TOKEN used to authenticate with Okta.

   c. Specify the Okta Filter to configure collection for events of your choosing. All events are collected by default unless you define an Okta API Filter expression for collecting the data, such as `filter=eventType eq "user.session.start".\n`. For Okta information to be weaved into authentication stories, "`user.authentication.sso`" events must be collected.

   d. Test the connection settings.

   e. If successful, Enable Okta log collection.

      Once events start to come in, a green check mark appears underneath the Okta configuration with the amount of data received.

6. After Cortex XSIAM begins receiving information from the service, you can Create an XQL Query to search for specific data. When including authentication events, you can also Create an Authentication Query to search for specific authentication data.

1.6.5 | Ingest endpoint data

Abstract

Cortex XSIAM enables you to ingest endpoint data.

Cortex XSIAM enables you to ingest endpoint data.

The following endpoint data can be ingested by Cortex XSIAM:

- SentinelOne DeepVisibility raw EDR events
- Microsoft Defender for Endpoint raw EDR events
- CrowdStrike Falcon Data Replicator raw EDR events
- CrowdStrike alerts and metadata, using CrowdStrike APIs
- Windows Events and other data using other Broker VM data collector applets

1.6.5.1 | Ingest alerts and metadata from CrowdStrike APIs

Abstract

Ingest CrowdStrike API real-time alerts and metadata for use in Cortex XSIAM stories.

> **NOTE:**
>
> To enable some of the APIs, you may need to reach out to CrowdStrike support.

To receive CrowdStrike API real-time alerts and logs, you must first configure data collection from CrowdStrike APIs. You can then configure the Data Sources settings in Cortex XSIAM for the CrowdStrike APIs.

> **NOTE:**
>
> For more information on configuring data collection from CrowdStrike APIs, see the CrowdStrike Documentation.

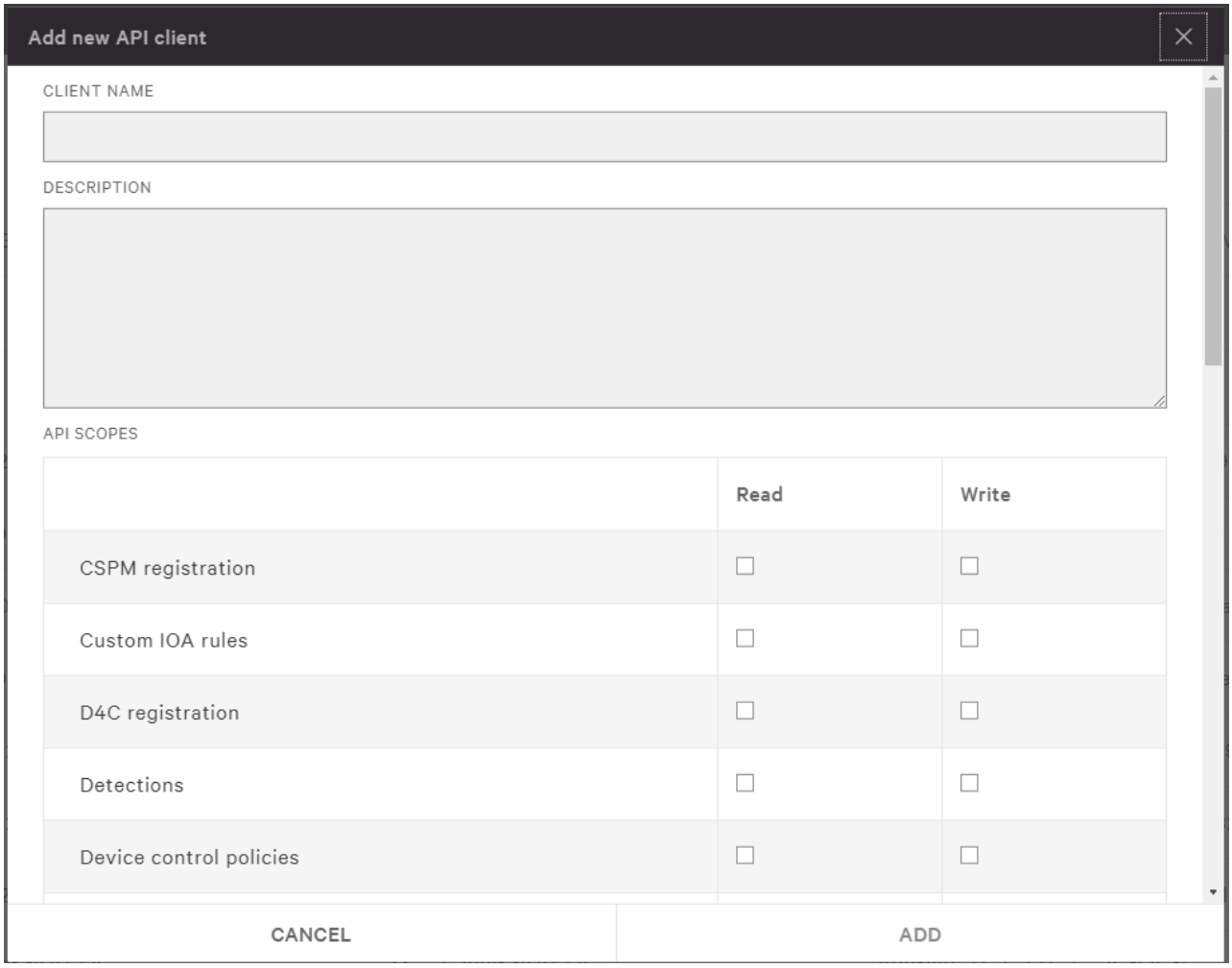
When Cortex XSIAM begins receiving alerts and logs, it automatically creates a CrowdStrike API XQL dataset (`crowdstrike_falcon_incident_raw`). You can use the issues created by Cortex XSIAM in rules, and search the logs using XQL Search. For example queries, refer to the in-app XQL Library.

1. Configure data collection from CrowdStrike APIs.

   a. In the CrowdStrike Falcon application, select  Support → API Clients and Keys.

   b. Under the OAuth2 API Clients section, Add new API client.

   c. Configure your new API client with these settings:

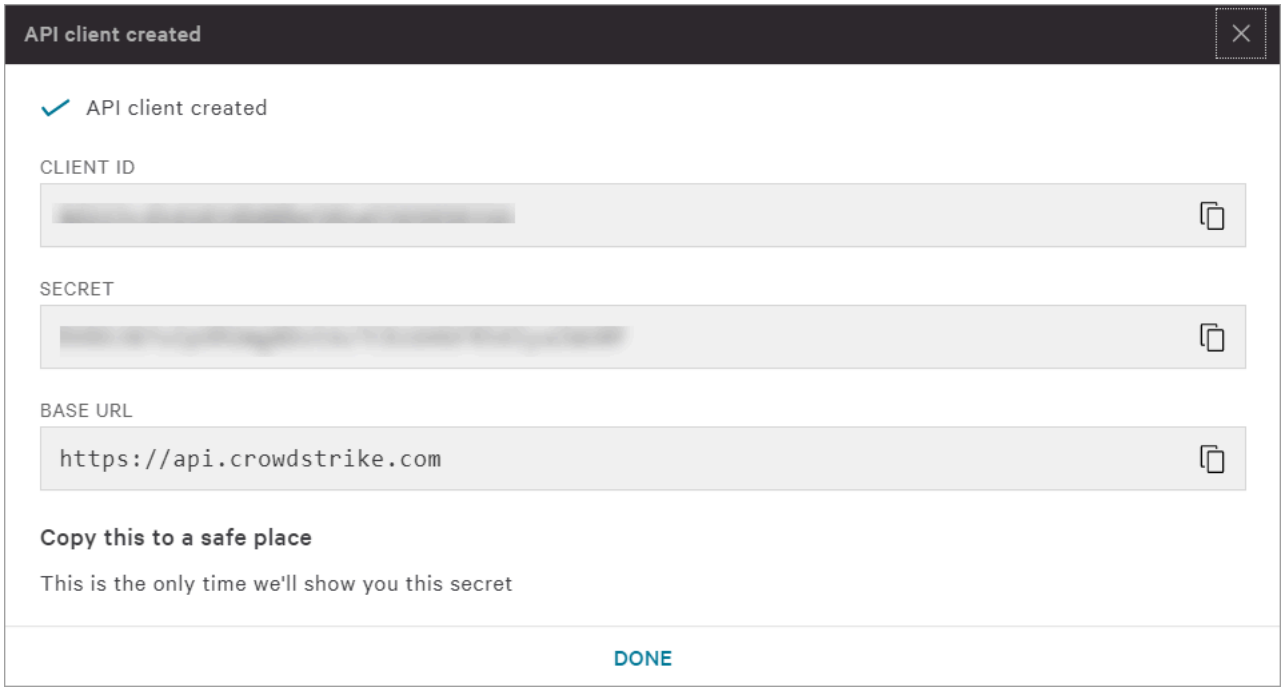


- CLIENT NAME: Specify a name for the new API client.
- DESCRIPTION: (Optional) Specify a description for the new API client.
- API SCOPES → Event streams: Select the Read permissions check box.
- API SCOPES → Hosts: Select the Read permissions check box.

d. Click ADD.

e. Copy the values for the CLIENT ID, SECRET, and BASE URL, and save them, because you will need them when you configure the Data Collection settings in Cortex XSIAM.

> **NOTE:**
>
> Ensure that you save the SECRET value because this is the only time that it is displayed.



f. Click DONE.

2. Configure the CrowdStrike Platform collection in Cortex XSIAM.

a. In Cortex XSIAM, select Settings → Data Sources.

b. On the Data Sources page, click Add Data Source, search for and select CrowdStrike Platform, and click Connect.

c. Set these parameters:

- Name: Specify a descriptive name for your log collection configuration, preferably the same CLIENT NAME used when adding a new client API in the CrowdStrike Falcon application, as explained above.
- Base URL: Specify the BASE URL you received when you created the client API in the CrowdStrike Falcon application, as explained above.
- Client ID: Specify the CLIENT ID you received when you created the client API in the CrowdStrike Falcon application, as explained above.
- Secret: Specify the SECRET you received when you created the client API in the CrowdStrike Falcon application, as explained above.
- Collect: Select the items that you want to collect (Alerts, Hosts).

d. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears below the CrowdStrike Platform configuration, along with the amount of data received.

1.6.5.2 | Ingest raw EDR events from CrowdStrike Falcon Data Replicator

Abstract

Ingest raw EDR event data from CrowdStrike Falcon Data Replicator into Cortex XSIAM.

Cortex XSIAM enables ingestion of raw EDR event data from CrowdStrike Falcon Data Replicator (FDR), streamed to Amazon S3. In addition to all standard SIEM capabilities, this integration unlocks some advanced Cortex XSIAM features, enabling comprehensive analysis of data from all sources, enhanced detection and response, and deeper visibility into CrowdStrike FDR data.

Key benefits include:

- Querying all raw event data received from CrowdStrike FDR using XQL.
- Querying critical modeled and unified EDR data via the `xdr_data` dataset.
- Enriching case and issue investigations with relevant context.
- Grouping issues with issues from other sources to accelerate the scoping process of cases, and to cut investigation time.
- Leveraging the data for analytics-based detection.
- Utilizing the data for rule-based detection, including correlation rules, BIOC, and IOC.
- Leveraging the data within playbooks for case response.

When Cortex XSIAM begins receiving EDR events from CrowdStrike FDR, it automatically creates a new dataset labeled `crowdstrike_fdr_raw`, allowing you to query all CrowdStrike FDR events using XQL. For example XQL queries, refer to the in-app XQL Library.

In addition, Cortex XSIAM parses and maps critical data into the `xdr_data` dataset and XDM data model, enabling unified querying and investigation across all supported EDR vendors' data, and unlocking key benefits like stitching and advanced analytics. While mapped data from all supported EDR vendors, including CrowdStrike, will be available in the `xdr_data` dataset, it's important to note that third-party EDR data present some limitations.

Third-party agents, including CrowdStrike, typically provide less data compared to our native agents, and do not include the same level of optimization for causality analysis and cloud-based analytics. Furthermore, external EDR rate limits and filters might restrict the availability of critical data required for comprehensive analytics. As a result, only a subset of our analytics-based detectors will function with third-party EDR data.

Raw event data from CrowdStrike FDR lacks key contextual information. To enhance its usability, we allocate additional resources to stitch it with other event data and data sources. Therefore, enabling the CrowdStrike FDR integration might temporarily make the tenant unavailable for a maintenance period of up to an hour.

We are continuously enhancing our support and using advanced techniques to enrich missing third-party data, while somehow replicating some proprietary functionalities available with our agents. This approach maximizes value for our customers using third-party EDRs within existing constraints. However, it’s important to recognize that the level of comprehensiveness achieved with our native agents cannot be matched, as much of the logic happens on the agent itself. These capabilities are unique, and are not found in typical SIEMs. Many of them, along with their underlying logic, are patented by Palo Alto Networks. Therefore, they should be regarded as added value beyond standard SIEM functionalities for customers who are not using our agents.

> **PREREQUISITE:**
>
> Ensure that your organization has a license for the CrowdStrike Falcon Data Replicator (FDR).
>
> Ensure that CrowdStrike FDR is enabled. CrowdStrike FDR can only be enabled by CrowdStrike Support. If CrowdStrike FDR is not enabled, submit a support ticket through the CrowdStrike support portal.
>
> Follow these steps to check if CrowdStrike FDR is enabled:
>
> 1. Log in to the CrowdStrike Falcon user interface using an account that has view/create permission for the API clients and keys page.
> 2. Navigate to Support → API Clients and Keys.
> 3. Verify that FDR AWS S3 Credentials and SQS Queue is listed.

> **NOTE:**
>
> Due to limitations with the S3 bucket used by CrowdStrike, data can only be collected once, by one system.

> **NOTE:**
>
> For more information on configuring data collection from CrowdStrike via Falcon Data Replicator, see CrowdStrike documentation.

Task 1: Create a CrowdStrike FDR feed

1. In the CrowdStrike user interface, select Support and resources → Resources and Tools → Falcon data replicator.
2. Click the FDR feeds tab.
3. Click Create feed.
4. Enter a feed name.
5. In Falcon Flight Control deployments, there is an option called Select which CID will manage this feed. In typical environments, the parent CID manages the feed for all of its child CIDs. This creates an aggregated feed that has data from all of the child CIDs. For information about aggregated feeds, and how they compare to individual feeds, see CrowdStrike documentation.
   - To set up an aggregated feed, select the parent CID.
   - To set up an individual feed, select a child CID or select both a parent CID and the Exclude Child CIDs option.
   - To exclude only some of the child CIDs, don’t select the Exclude Child CIDs option. Instead, select Customize your FDR feed in the next step.
6. Set the feed status.

7. Select the method for creating your feed, from the following options:
    - Create your FDR feed with default settings, where you get the recommended settings, including all current and future events, all secondary events (if available), and no partitions.
    - Customize your FDR feed, where you start with the option to use a filter to get the specific events that you want in the feed. You can then customize secondary events and partitioning.
8. Include secondary events. They are required for data stitching and enrichment.
9. Optionally, in Flight Control deployments, edit the existing child CIDs included in the feed, and choose whether future CIDs are automatically included, by using the Include future CIDs option.
10. Click Create feed.
11. From the summary page that appears, copy and save all the information shown on the page somewhere safe, for later use. This page includes the credentials that are required for setting up an SQS consumer.

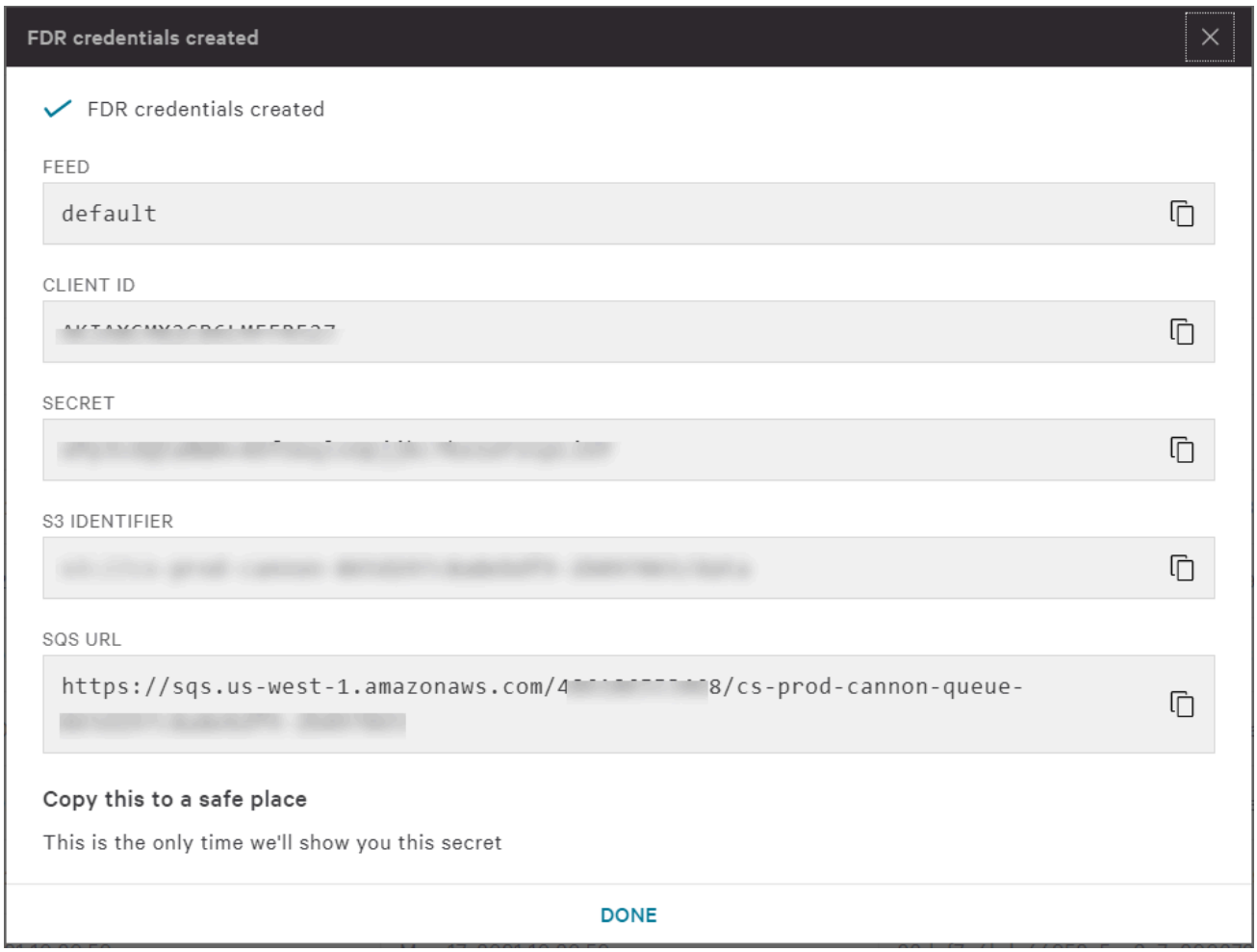
> **NOTE:**
>
> Ensure that you copy the Secret, and store it in a safe place. You will not be able to retrieve it later. If you need a new secret, you must reset the feed credentials.

Task 2: Configure CrowdStrike Falcon Data Replicator

1. Log in to CrowdStrike Falcon using an account that has view/create permission for the API clients and keys page.
2. Navigate to  Support → API Clients and Keys.
3. On the same line as FDR AWS S3 Credentials and SQS Queue, click Create new credentials.

> **NOTE:**
>
> CrowdStrike Falcon Data Replicator only supports one FDR credential configuration.

4. Configure your new FDR credentials.



5. Copy the values for the CLIENT ID, SECRET, S3 IDENTIFIER, and SQS URL, and save them somewhere safe, because you will need them when you configure data collection in Cortex XSIAM.

   > **NOTE:**
   >
   > Ensure that you save the SECRET value, because this is the only time that it is displayed. You can go back to this page later to copy the other credentials, but you will not have access to the secret again.

6. Click DONE.

Task 3: Configure ingestion into Cortex XSIAM

1. In Cortex XSIAM, select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select CrowdStrike Falcon Data Replicator, and click Connect.
3. Set these parameters:
   - Name: Specify a descriptive name for your log collection configuration.
   - SQS URL: Specify the SQS URL you received when you created the FDR credential in CrowdStrike Falcon, as explained above.
   - AWS Client ID: Specify the CLIENT ID you received when you created the FDR credential in CrowdStrike Falcon, as explained above.
   - AWS Client Secret: Specify the SECRET you received when you created the FDR credential in CrowdStrike Falcon, as explained above.
4. Click Test to validate access, and then click Enable.

When events start to come in, a green check mark appears below the CrowdStrike Falcon Data Replicator configuration, along with the amount of data received.

1.6.5.3 | Ingest raw EDR events from Microsoft Defender for Endpoint

Abstract

Ingest raw EDR event data from Microsoft Defender for Endpoint Events into Cortex XSIAM.

Cortex XSIAM enables ingestion of raw EDR event data from Microsoft Defender for Endpoint Events, streamed to Azure Event Hubs. In addition to all standard SIEM capabilities, this integration unlocks some advanced Cortex XSIAM features, enabling comprehensive analysis of data from all sources, enhanced detection and response, and deeper visibility into Microsoft Defender for Endpoint data.

Key benefits include:

- Querying all raw event data received from Microsoft Defender for Endpoint using XQL.
- Querying critical modeled and unified EDR data via the `xdr_data` dataset.
- Enriching case and issue investigations with relevant context.
- Grouping issues with issues from other sources to accelerate the scoping process of cases, and to cut investigation time.
- Leveraging the data for analytics-based detection.
- Utilizing the data for rule-based detection, including correlation rules, BIOC, and IOC.
- Leveraging the data within playbooks for case response.

When Cortex XSIAM begins receiving EDR events from Microsoft Defender for Endpoint Events, it automatically creates a new dataset labeled `msft_defender_raw`, allowing you to query all Microsoft Defender for Endpoint Events using XQL. For example XQL queries, refer to the in-app XQL Library.

In addition, Cortex XSIAM parses and maps critical data into the `xdr_data` dataset and XDM data model, enabling unified querying and investigation across all supported EDR vendors' data, and unlocking key benefits like stitching and advanced analytics. While mapped data from all supported EDR vendors, including Microsoft Defender for Endpoint Events, will be available in the `xdr_data` dataset, it's important to note that third-party EDR data present some limitations.

Third-party agents, including Microsoft Defender for Endpoint Events, typically provide less data compared to our native agents, and do not include the same level of optimization for causality analysis and cloud-based analytics. Furthermore, external EDR rate limits and filters might restrict the availability of critical data required for comprehensive analytics. As a result, only a subset of our analytics-based detectors will function with third-party EDR data.

We are continuously enhancing our support and using advanced techniques to enrich missing third-party data, while somehow replicating some proprietary functionalities available with our agents. This approach maximizes value for our customers using third-party EDRs within existing constraints. However, it’s important to recognize that the level of comprehensiveness achieved with our native agents cannot be matched, as much of the logic happens on the agent itself. These capabilities are unique, and are not found in typical SIEMs. Many of them, along with their underlying logic, are patented by Palo Alto Networks. Therefore, they should be regarded as added value beyond standard SIEM functionalities for customers who are not using our agents.

> **NOTE:**

The generic Cortex XSIAM Azure Event Hub collector does not offer full functionality for EDR data (such as stitching), and is therefore not suitable for EDR data ingestion.

Task 1: Configure Microsoft Defender for Endpoint Events to stream raw data to Microsoft Azure Event Hub

PREREQUISITE:

Ensure that you do the following tasks before you begin configuring data collection.

- Create an Azure Event Hub. For more information, see Quickstart: Create an event hub using Azure portal.
    1. Create a resource group (optional if you already have a resource group configured).
    2. Create an Event Hubs namespace.
    3. Create an event hub within the namespace. On the Settings → Networking page → Public Access tab, ensure that you add Palo Alto Networks IP addresses to the Firewall allow list. Set Exception to Yes.
    4. Ensure that you keep a copy of the Event Hub resource ID and the Event Hub name for use in the following procedures. To get your Event Hubs resource ID, go to your Azure Event Hub namespace page on Azure's Properties tab, and copy the text under Resource ID.
    5. Create a storage account.
- Ensure that you have Microsoft Defender user credentials to sign in as a Security Administrator.

1. Enable raw data streaming:
    a. Sign in to the Microsoft Defender portal as a Security Administrator.
    b. Go to the data export settings page in the Microsoft Defender portal: System → Settings → Windows Defender XDR → Streaming API.
    c. Click +Add.
    d. In the Name box, enter a name for your new data streaming settings.
    e. Select Forward events to Event Hub.
    f. In the Event-Hub Resource ID box, enter the Event Hub resource ID that you prepared in advance.
    g. In the Event-Hub box, enter the Event Hub name that you prepared in advance.
    h. For Event Types, select the event types that you want to stream.

    NOTE:

    If you select all event types and leave Event-Hub name empty, an event hub will be created for each category in the selected namespace. If you are not using a Dedicated Event Hubs ClusterEvent Hub, namespaces have a limit of 10 Event Hubs.

    i. Click Submit.
    j. Verify that the events that you selected are streaming by going to your Event Hubs namespace, Settings → Networking. Select the Event Hub name and the Consumer group, and then under Advanced properties, click View events. Check the Event body.
2. In the Microsoft Azure console, open the Event Hubs page, and select the Azure Event Hub that you created for collection of Microsoft Defender logs.
3. Save a copy of the following parameters from your configured event hub, because you will need them when configuring data collection in Cortex XSIAM:
    - Your event hub's consumer group:
        1. Select Entities → Event Hubs, and select your event hub.
        2. Select Entities → Consumer groups, and select your event hub.
        3. In the Consumer group table, copy the applicable value listed in the Name column for your Cortex XSIAM data collection configuration.
    - Your event hub's connection string for the designated policy:
        1. Select Settings → Shared access policies.
        2. In the Shared access policies table, select the applicable policy.
        3. Copy the Connection string-primary key.
    - Your storage account connection string required for partitions lease management and checkpointing in Cortex XSIAM:
        1. Open the Storage accounts page, and either create a new storage account or select an existing one, which will contain the storage account connection string.
        2. Select Security + networking → Access keys, and click Show keys.
        3. Copy the applicable Connection string.

Task 2: Configure the Microsoft Defender for Endpoint Events collector in Cortex XSIAM

1. Select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select Microsoft Defender for Endpoint, and click Connect.
3. Set these parameters:
    - Name: Specify a unique descriptive name for your log collection configuration. You cannot change this name later.
    - Event Hub Connection String: Specify your event hub’s connection string for the designated policy.
    - Storage Account Connection String: Specify your storage account’s connection string for the designated policy.
    - Consumer Group: Specify your event hub’s consumer group.
4. Click Test to validate access, and then click Save.

    When events start to come in, a green check mark appears beneath the Microsoft Defender for Endpoint configuration, with the amount of data received.

1.6.5.4 | Ingest raw EDR events from SentinelOne DeepVisibility

Abstract

Ingest raw EDR event data from SentinelOne DeepVisibility into Cortex XSIAM.

Cortex XSIAM enables ingestion of raw EDR event data from SentinelOne DeepVisibility, streamed via Cloud Funnel to Amazon S3. In addition to all standard SIEM capabilities, this integration unlocks some advanced Cortex XSIAM features, enabling comprehensive analysis of data from all sources, enhanced detection and response, and deeper visibility into SentinelOne data.

Key benefits include:

- Querying all raw event data received from SentinelOne using XQL.
- Querying critical modeled and unified EDR data via the `xdr_data` dataset.
- Enriching case and issue investigations with relevant context.
- Grouping issues with issues from other sources to accelerate the scoping process of cases, and to cut investigation time.
- Leveraging the data for analytics-based detection.
- Utilizing the data for rule-based detection, including correlation rules, BIOC, and IOC.
- Leveraging the data within playbooks for case response.

When Cortex XSIAM begins receiving EDR events from SentinelOne, it automatically creates a new dataset labeled `sentinelone_deep_visibility_raw`, allowing you to query all SentinelOne events using XQL. For example XQL queries, refer to the in-app XQL Library.

In addition, Cortex XSIAM parses and maps critical data into the `xdr_data` dataset and XDM data model, enabling unified querying and investigation across all supported EDR vendors' data and unlocking key benefits like stitching and advanced analytics. While mapped data from all supported EDR vendors, including SentinelOne DeepVisibility, will be available in the `xdr_data` dataset, it's important to note that third-party EDR data present some limitations.

Third-party agents, including SentinelOne, typically provide less data compared to our native agents, and do not include the same level of optimization for causality analysis and cloud-based analytics. Furthermore, external EDR rate limits and filters might restrict the availability of critical data required for comprehensive analytics. As a result, only a subset of our analytics-based detectors will function with third-party EDR data.

We are continuously enhancing our support and using advanced techniques to enrich missing third-party data, while somehow replicating some proprietary functionalities available with our agents. This approach maximizes value for our customers using third-party EDRs within existing constraints. However, it’s important to recognize that the level of comprehensiveness achieved with our native agents cannot be matched, as much of the logic happens on the agent itself. These capabilities are unique, and are not found in typical SIEMs. Many of them, along with their underlying logic, are patented by Palo Alto Networks. Therefore, they should be regarded as added value beyond standard SIEM functionalities for customers who are not using our agents.

> **PREREQUISITE:**

- The SentinelOne DeepVisibility logs that will be collected by your dedicated Amazon S3 bucket must adhere to the following guidelines:
  - Each log file must use the 1 log per line format as multi-line format is not supported.
  - The log format must be compressed as gzip or uncompressed.
  - For best performance, we recommend limiting each file size to up to 50 MB (compressed).
- The minimum AWS permissions required for an Amazon S3 bucket and Amazon Simple Queue Service (SQS) are:
  - **Amazon S3 bucket**: `GetObject`
  - **SQS**: `ChangeMessageVisibility`, `ReceiveMessage`, and `DeleteMessage`
- Determine how you want to provide access to Cortex XSIAM to your logs and to perform API operations. You have the following options:
  - Designate an AWS IAM user, where you will need to know the Account ID for the user and have the relevant permissions to create an access key/id for the relevant IAM user. If you do not have a designated AWS IAM user configured yet, instructions for this are included in the following procedures.
  - Create an assumed role in AWS to delegate permissions to a Cortex XSIAM AWS service. This role grants Cortex XSIAM access to your flow logs. This is the Assumed Role option mentioned later in the procedures that follow. To create an assumed role for Cortex XSIAM, see Create an assumed role.

    For more information about assumed roles, see Creating a role to delegate permissions to an AWS service.
- To collect Amazon S3 logs that use server-side encryption (SSE), the user role must have an IAM policy that states that Cortex XSIAM has kms:Decrypt permissions. With this permission, Amazon S3 automatically detects if a bucket is encrypted and decrypts it. If you want to collect encrypted logs from different accounts, you must have the decrypt permissions for the user role also in the key policy for the master account Key Management Service (KMS). For more information, see Allowing users in other accounts to use a KMS key.

Task 1: Configure an Amazon S3 bucket

Task A: Create a dedicated Amazon S3 bucket to store SentinelOne DeepVisibility EDR data

This step provides general guidelines. For more information, see Creating a bucket using the Amazon S3 Console.

NOTE:

It is your responsibility to define a retention policy for your Amazon S3 bucket by creating a Lifecycle rule on the Management tab. We recommend setting the retention policy to at least 7 days to ensure that the data is retrieved under all circumstances.

1. Log in to the AWS Management Console and navigate to the S3 Service.
2. Create a new S3 bucket:
   a. Click Create bucket.
   b. For Bucket Name, enter a unique name for the bucket (for example, `xsiam-s1-edr-data`).
   c. Choose an appropriate AWS Region.
   d. Set Block all public access to Enabled.
   e. Click Create bucket.
3. Set up the Bucket policy:
   a. Click the Permissions tab of your new bucket.
   b. Under Bucket policy, click Edit and add the following policy to allow SentinelOne DeepVisibility to write data there.

      Replace `your-sentinelone-account-id` with the relevant value for your environment; replace `xsiam-s1-edr-data` with the name of your new bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "your-sentinelone-account-id"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::xsiam-s1-edr-data/*"
        }
    ]
}
```

Task B: Configure an Amazon Simple Queue Service (SQS) and grant it permission to receive messages from S3

NOTE:

Ensure that you create your Amazon S3 bucket and Amazon SQS queue in the same region.

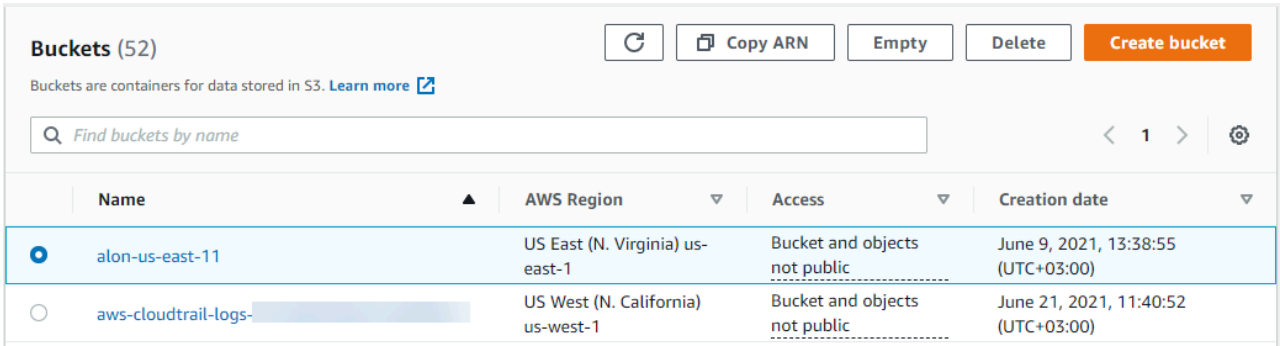
1. In the Amazon SQS Console, click Create Queue.

2. Configure the following settings, where the default settings should be configured unless otherwise indicated.

   - Type: Select Standard queue (default).
   - Name: Specify a descriptive name for your SQS queue.
   - Configuration section: Keep the default settings for the various fields.
   - Access policy → Choose method: Select Advanced and update the Access policy code in the editor window to enable your Amazon S3 bucket to publish event notification messages to your SQS queue. Use this sample code as a guide for defining the "`Statement`" with the following definitions.

     "`Resource`": Keep the automatically generated ARN for the SQS queue that is set in the code, which uses the format "`arn:sns:Region:account-id:topic-name`".

     You can retrieve your bucket's ARN by opening the Amazon S3 Console in a browser window. In the Buckets section, select the bucket that you created for collecting the Amazon S3 flow logs, click Copy ARN, and paste the ARN in the field.



   > **NOTE:**
   >
   > For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    }
  ]
}
```

   - Dead-letter queue section: We recommend that you configure a queue for sending undeliverable messages by selecting Enabled, and then in the Choose queue field selecting the queue to send the messages. You may need to create a new queue for this, if you do not already have one set up. For more information, see Amazon SQS dead-letter queues.

3. Click Create queue.

When the SQS is created, a message indicating that the queue was successfully configured is displayed at the top of the page.

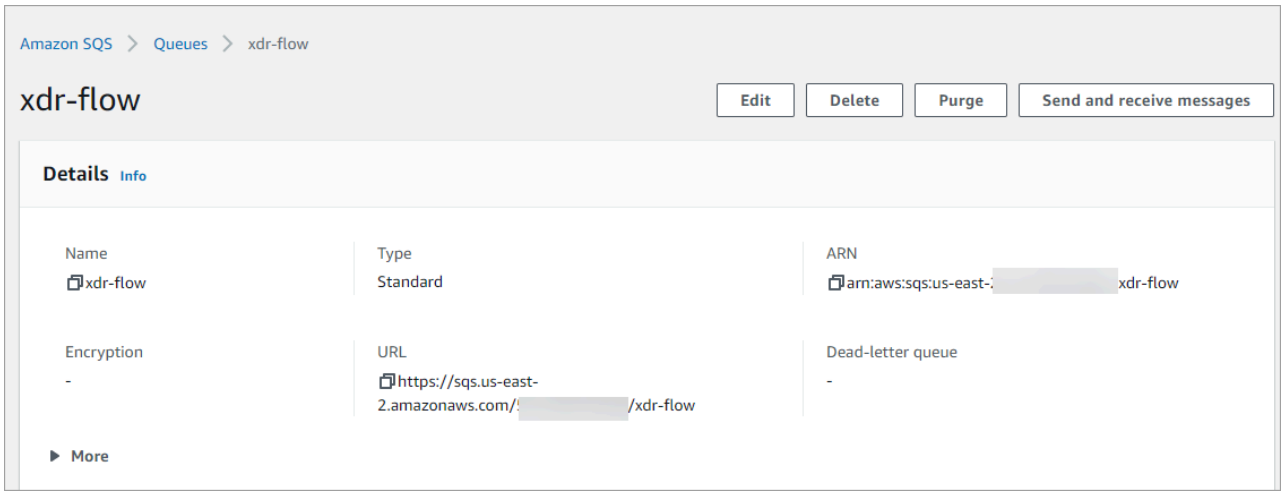
Task C: Configure an event notification to your Amazon SQS whenever a file is written to your Amazon S3 bucket

1. Open the Amazon S3 Console and in the Properties tab of your Amazon S3 bucket, scroll down to the Event notifications section, and click Create event notification.

2. Configure the following settings:

- Event name: Specify a descriptive name for your event notification containing up to 255 characters.
- Prefix: Do not set a prefix, because the Amazon S3 bucket is meant to be a dedicated bucket for collecting only network flow logs.
- Event types: Select All object create events for the type of event notifications that you want to receive.
- Destination: Select SQS queue to send notifications to an SQS queue to be read by a server.
- Specify SQS queue: You can either select Choose from your SQS queues and then select the SQS queue, or select Enter SQS queue ARN and specify the ARN in the SQS queue field.

  You can retrieve your SQS queue ARN by opening another instance of the AWS Management Console in a browser window, opening the Amazon SQS Console, and selecting the Amazon SQS that you created. In the Details section, under ARN, click the copy icon (⧉), and paste the ARN in the field.



3. Click Save changes.

   When the event notification is created, a message indicating that the event notification was successfully created is displayed at the top of the page.

> **NOTE:**
>
> If you receive an error when trying to save your changes, check that the permissions are set up correctly, and fix them if necessary.

Task D: Configure authentication\authorization if you have not done so yet

For Assumed Role, follow these instructions: Create an assumed role, and then return to this page to Configure SentinelOne DeepVisibility.

For IAM access key:

1. Create an IAM Policy that grants permissions for SQS and S3:

   a. In the AWS Console, navigate to the IAM service, and click Policies.

   b. Click Create policy.

   c. Select the JSON policy editor.

   d. Use this sample code as a guide for defining the “Statement” with the following definitions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ReadAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "[ARN for the S3 Bucket Name defined by AWS]", #example: "arn:aws:s3:::bucketname/xsiam-s1-edr-data/"
                "[ARN for the S3 Bucket path defined by AWS]" #example: "arn:aws:s3:::bucketname/xsiam-s1-edr-data/*"
            ]
        },
        {
            "Sid": "SQSReceiveAccess",
            "Effect": "Allow",
            "Action": [
                "sqs:ReceiveMessage",
                "sqs:GetQueueAttributes"
            ],
            "Resource": "[ARN for the SQS queue defined by AWS]"
        }
```

```
    ]
}
```

e. Click Next.

f. For Policy name, enter a name.

g. Click Create policy.

2. Create an IAM User:

a. In the AWS Console, navigate to the IAM service, and click Users.

b. Click Create user.

c. For User name, enter a name (for example, `cortex-xsiam-s3`).

d. Attach the IAM Policy that you created in Step 1.

e. Click Next.

f. Click Create user.

3. Configure access keys for the AWS IAM User:

> **NOTE:**
>
> It is the responsibility of your organization to ensure that the user who creates the access key is assigned the relevant permissions. Otherwise, this can cause the process to fail with errors.

a. Open the AWS IAM Console, and in the navigation pane, select Access management → Users.

b. Select the User name of the AWS IAM user.

c. Select the Security credentials tab, scroll down to the Access keys section, and click Create access key.

d. Click the copy icon next to the Access key ID and Secret access key keys, where you must click Show secret access key to see the secret key, and save a copy of them somewhere safe before closing the window. You will need to provide these keys when you edit the Access policy of the SQS queue, and when setting the AWS Client ID and AWS Client Secret in Cortex XSIAM. If you forget to record the keys and close the window, you will need to generate new keys and repeat this process.

> **NOTE:**
>
> For more information, see Managing access keys for IAM users.

4. Update the Access policy of your Amazon SQS queue:

a. In the Amazon SQS Console, select the SQS queue that you created when you configured an Amazon Simple Queue Service (SQS).

b. Select the Access policy tab, and click Edit to edit the Access policy code in the editor window, to enable the IAM user to perform operations on the Amazon SQS with the permissions `SQS:ChangeMessageVisibility`, `SQS:DeleteMessage`, and `SQS:ReceiveMessage`. Use this sample code as a guide for defining the “`Sid”: “__receiver_statement`” with the following definitions.

- “aws:SourceArn”: Specify the ARN of the AWS IAM user. You can retrieve the User ARN from the Security credentials tab, which you accessed when you configured access keys for the AWS API user.
- “Resource”: Keep the automatically generated ARN for the SQS queue that is set in the code, which uses the format “arn:sns:Region:account-id:topic-name”.

> **NOTE:**
>
> For more information on granting permissions to publish messages to an SQS queue, see Granting permissions to publish event notification messages to a destination.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "[ARN of your Amazon S3 bucket]"
        }
      }
    },
    {
      "Sid": "__receiver_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "[Add the ARN for the AWS IAM user]"
      },
      "Action": [
        "SQS:ChangeMessageVisibility",
        "SQS:DeleteMessage",
        "SQS:ReceiveMessage"
      ],
      "Resource": "[Leave automatically generated ARN for the SQS queue defined by AWS]"
    }
  ]
}
```

c. Click Save.

Task 2: Configure SentinelOne DeepVisibility

1. In SentinelOne DeepVisibility, select Configure → Policy & Settings and in the Singularity Data Lake section, click Cloud Funnel.
2. For Cloud Provider, select AWS (Amazon Web Services).
3. For S3 Bucket Name, enter the name of the Amazon S3 bucket that you created for SentinelOne DeepVisibility log ingestion.
4. For Telemetry Streaming, select Enable.
5. In the Query Filters box, create a query that includes the agents that should send data to the S3 bucket.
6. To validate the query, click Validate.
7. For Fields to include, ensure that all fields are selected.
8. Click Save.

Task 3: Configure ingestion into Cortex XSIAM

1. In Cortex XSIAM, select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select `SentinelOne - Deep Visibility`, and click Connect.
3. Use the toggle to select either Access Key or Assumed Role.
4. Set these parameters, depending on your choice in the previous step:

- For the Access Key option:
    - Name: Specify a descriptive name for your log collection configuration. This name must be unique in your environment.
    - SQS URL: Specify the SQS URL that you received for the AWS S3 queue when you configured the Amazon Simple Queue Service (SQS), as explained above.
    - AWS Client ID: Specify the Client ID that you received when you configured the AWS IAM user, as explained above.
    - AWS Client Secret: Specify the Secret that you received when you configured the AWS IAM user, as explained above.
- For the Assumed Role option:
    - Name: Specify a descriptive name for your log collection configuration. This name must be unique in your environment.
    - SQS URL: Specify the SQS URL that you received for the AWS S3 queue when you configured the Amazon Simple Queue Service (SQS), as explained above.
    - Role ARN: Specify the role ARN that you received when you created the assumed role.
    - External Id: Specify the External ID that you received when you created the assumed role.

5. Click Test to validate access, and then click Enable.

   After events start to come in, a green check mark appears below the SentinelOne - DeepVisibility configuration, along with the amount of data received.

1.6.6 | Ingest cloud assets

Abstract

Explains how to onboard cloud service providers from the Data Source page.

Cortex XSIAM provides a unified, normalized asset inventory for cloud assets. This capability provides deeper visibility to all the assets and superior context for incident investigation.

The cloud service provider (CSP) onboarding wizard is designed to facilitate the seamless setup of CSP data into Cortex XSIAM. The guided experience requires minimal user input; simply define the scope of your CSP accounts and specify the scan mode. For full control of the CSP setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM generates an authentication template to establish trust to the CSP and grant permissions to Cortex XSIAM. The template must be executed in the CSP to complete the onboarding process. Execution of the template grants the permissions and includes a component that notifies Cortex XSIAM of the execution details and a new cloud instance is created.

> **NOTE:**
>
> The cloud accounts being onboarded must be owned by the customer performing the onboarding process.

You can leverage your CSP hierarchy and choose whether to onboard individual accounts one at a time or collection of accounts (such as organization in AWS and GCP or management group in Azure). Various options are available for each CSP to allow you to customize your data collection.

Cortex XSIAM supports two scan modes:

- Cloud scan: (Recommended) The scanning takes place within the Cortex XSIAM cloud environment. No additional setup is needed.
- Outpost scan: The scanning is performed on infrastructure deployed to a CSP account owned by you. The CSP account should be a dedicated account for the outpost, free from other resources. Each CSP account can host only one outpost. This mode requires additional cloud provider permissions and may incur additional cloud costs.

To allow you to fine tune your CSP data collection, you can modify the scope of data collection by including or excluding specific regions. If you selected to collect data from an organizational unit that is not the lowest on the CSP hierarchy (such as organization or organizational unit in AWS, organization or folder in GCP, and tenant or management group in Azure), you can also modify the scope by including or excluding specific accounts, projects, or subscriptions. If you choose to include specific accounts, only those specified accounts will be included, even if additional accounts are added to the CSP after onboarding. If you choose to exclude specific accounts, any new accounts added to the CSP after onboarding will be included in the scope. Excluded accounts are not visible in Cortex XSIAM.

The advanced settings allow you to select which Cortex XSIAM modules you want to enable for this CSP. By default, the following security capabilities are enabled:

- Discovery engine
- Cloud security posture management
- Cloud infrastructure entitlement management
- Agentless disk scanning
- AI security posture management

The additional security capabilities you can enable include:

- XSIAM analytics: Analyzes your endpoint data to develop a baseline and raise Analytics and Analytics BIOC alerts when anomalies and malicious behaviors are detected.
- Data security posture management: An agentless multi-cloud data security solution that discovers, classifies, protects, and governs sensitive data.
- Registry scanning: Scan container registry images for vulnerabilities. malware, and secrets. You can configure your initial preference for scanning your registry. Any newly discovered registry, repository or image in the account will be scanned by default.

1.6.6.1 | Onboard Amazon Web Services

Abstract

Follow the AWS onboarding wizard and Cortex creates a custom authentication template to be executed in AWS.

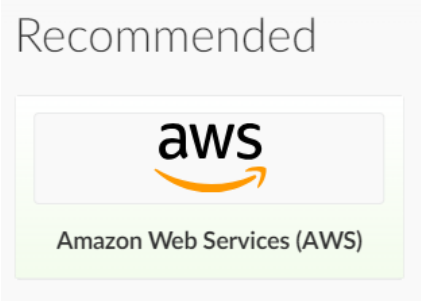
> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on.

Follow this wizard to onboard your Amazon Web Services (AWS) environment. The AWS onboarding wizard is designed to facilitate the seamless setup of AWS data into Cortex XSIAM. The guided experience requires minimal user input; simply define the scope of your AWS accounts and specify the scan mode. For full control of the setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM generates an authentication template to establish trust to AWS and grant permissions to Cortex XSIAM. The template must be executed in AWS to complete the onboarding process. Execution of the template grants the permissions and includes a component that notifies Cortex XSIAM of the execution details and a new cloud instance is created.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - Access to AWS Management Console
> - Required AWS permissions

How to onboard AWS:

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select Amazon Web Services (AWS) and click Connect.



4. In the onboarding wizard, select the scope for this data source:
   - Organization: (Default) A collection of AWS accounts that are managed centrally.
   - Organizational Unit: A group of AWS accounts within an organization. It can also contain other organizational units.
   - Account: A specific AWS member account.
5. Choose the Scan Mode:
   - Cloud Scan: (Recommended) Security scanning is performed in the Cortex cloud environment.
   - Scan with Outpost: Security scanning is performed on infrastructure deployed to a cloud account owned by you. If you select this option, choose the outpost account to use for this instance.

     > **NOTE:**
     >
     > Scanning with an outpost may require additional CSP permissions and may incur additional CSP costs.
6. (Optional) Click Show advanced settings to define advanced settings:

- Instance Name: Enter a unique instance name or leave empty to be automatically populated. The automatic naming convention is the CSP name followed by the ID of the scope unit selected in the onboarding wizard. For example, when onboarding an Amazon Web Services account, the automatic name would be `AWS-<accountID>` where `<accountID>` is the ID of the account onboarded.
- Scope Modifications: To allow you to fine tune your AWS scope, you can modify the scope by including or excluding specific regions. Additionally, if you selected organization or organizational unit as the scope, you can modify the scope by including or excluding specific accounts. If you choose to include specific accounts, only those specified accounts will be included, even if additional accounts are added to your AWS environment after onboarding. If you choose to exclude specific accounts, any new accounts added to your AWS environment after onboarding will be included in the scope.

  > **NOTE:**
  >
  > When onboarding an AWS organization or organizational unit (OU), Cortex XSIAM creates IAM resources in every account within that organization or OU. This occurs even if you choose to exclude specific accounts from being scanned. While excluded accounts will not be scanned and will not appear in the asset inventory, the IAM resources may still be present.

- Additional Security Capabilities: Choose from which security capabilities you want to benefit. Some security capabilities are enabled by default and can be modified. Adding security capability typically requires additional cloud provider permissions. For detailed information on the permissions required, see Cloud service provider permissions. The additional security capabilities you can enable include:
  - XSIAM analytics: Analyzes your endpoint data to develop a baseline and raise Analytics and Analytics BIOC alerts when anomalies and malicious behaviors are detected.
  - Data security posture management: An agentless data security scanner that discovers, classifies, protects, and governs sensitive data.
  - Registry scanning: A container registry scanner that scans registry images for vulnerabilities. malware, and secrets. For more details, see Configure registry scanning
  - Serverless functions scanning: Implement serverless scanning to detect and remediate vulnerabilities within serverless functions during the development lifecycle. Seamless integration into CI/CD pipelines enables automated security scans for a continuously secure pre-production environment.

    > **NOTE:**
    >
    > See the prerequisites above for specific permissions required for serverless functions.

  - Automation: Pre-configures a list of integrations and associated commands to automate security issue responses. Commands can be utilized individually or as part of custom playbooks for issue remediation.
  - Agentless disk scanning: (Recommended) Implement agentless disk scanning to remotely detect and remediate vulnerabilities during the development lifecycle.
- Log Level: (Optional - for Automation only) Configure the automation integration logging level. Possible values are:
  - Off (Default)
  - Debug
  - Verbose
- Cloud Tags: Define tags and tag values to be added to any new resource created by Cortex in the cloud environment.
- Log Collection Configuration: To maximize security coverage, include collection of audit logs using CloudTrail. This may require additional cloud service provider permissions. For detailed information on the permissions required, see Cloud service provider permissions.

  For the purpose of collecting audit logs, the AWS onboarding wizard automatically provisions dedicated AWS resources in your AWS environment, specifically an AWS CloudTrail trail, Amazon SQS queue, and an Amazon S3 bucket. As a result, you may incur increased AWS costs, primarily due to CloudTrail event logging. While the trail defaults to capturing both read and write management events, the majority of these costs are typically associated specifically with read management events.

  To help manage these costs, you may manually modify the trail (`cortex-trail-<aws_account_id>`) configuration in the AWS Management Console to disable read events. While this reduces detection coverage, it should significantly lower CloudTrail-related charges. It is important to note that these manual changes will be overwritten during future Cortex XSIAM updates, but they can serve as a temporary measure for cost control.

7. Cortex XSIAM creates an instance in pending state.

8. To complete the process, execute the template in AWS using one of the following methods:

   - Automated: (Recommended) Click Execute in AWS to connect to AWS CloudFormation and create the stack.

     > **NOTE:**
     >
     > If you select Automated, you must already be logged in to AWS.

   - Manual: Click Download CloudFormation and follow the instructions to manually execute the CloudFormation template file in AWS CloudFormation.

> **NOTE:**
>
> The template is reusable and can be executed as many times as you want to create new instances with the settings you defined in the wizard.

9. Click Close.

   When the template is successfully uploaded to AWS and the stack creation is complete, a new instance is created and the initial discovery scan is started. When the scan is complete, you can view the discovered assets in Asset Inventory.

> **NOTE:**
>
> You can see the automatically created automation instance in the Automation & Feed Integrations page under the Cloud Services section, and it will have the same name as the cloud integration instance. The instance is read-only, you can only edit the instance from the Data Sources page.
>
> (AWS only) instances that were previously created in the Automation & Feed Integration page can also be edited or deleted in the Automation & Feed Integration page.

Required AWS permissions

To onboard AWS to Cortex XSIAM, the following IAM policy actions are required:

- iam:GetRole
- iam:UpdateAssumeRolePolicy
- iam:GetPolicyVersion
- iam:GetPolicy
- iam:UpdateRoleDescription
- iam:DeletePolicy
- iam:ListRoles
- iam:CreateRole
- iam:DeleteRole
- iam:AttachRolePolicy
- iam:PutRolePolicy
- iam:CreatePolicy
- iam:PassRole
- iam:CreateServiceLinkedRole
- iam:DetachRolePolicy
- iam:ListPolicyVersions
- iam:DeleteRolePolicy
- iam:UpdateRole
- iam:DeleteServiceLinkedRole
- iam:ListRolePolicies
- iam:GetRolePolicy
- iam:DeletePolicyVersion
- iam:SetDefaultPolicyVersion
- lambda:CreateFunction
- lambda:UpdateFunctionCode
- lambda:UpdateFunctionConfiguration
- lambda:GetFunction

Additional permissions are required for collecting audit logs and could be scoped to Cortex XSIAM created resources. These include the following types of permissions: KMS, S3, SQS , SNS , and Cloudtrail.

To enable serverless function scanning, grant the following permissions in your AWS account for scanning outposts and accessing logs:

```
{
 "Version": "2012-10-17",
```

```
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "lambda:GetFunction",
       "lambda:GetFunctionConfiguration",
       "lambda:GetLayerVersion",
       "iam:GetRole"
     ],
     "Resource": "*"
   }
 ]
}
```

1.6.6.1.1 | Manually upload template to AWS

Abstract

Learn how to manually create a stack in AWS Management Console using the CloudFormation file downloaded in the onboarding wizard.

When you have downloaded the CloudFormation template file in the onboarding wizard, you must connect to AWS Management Console to create a stack using the template file.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - An AWS account
> - Access to AWS Management Console
> - Permission to create a stack and its resources in AWS CloudFormation

1. In AWS Management Console, navigate to CloudFormation.
2. On the Stacks page, click Create stack, and then select With new resources (standard).
3. On the Create stack page, in Prerequisite - Prepare template, select Choose an existing template.
4. In Specify template, select Upload a template file, then click Choose file and upload the template downloaded from your Cortex Platform. Click Next.
5. In the Specify stack details page, enter a Stack name.
6. In Parameters, enter a unique Amazon Resource Name (ARN) for the custom CortexPrismaRoleName role, and an ExternalID. Click Next and Next again.
7. In Review, acknowledge that CloudFormation might create IAM resources with custom names and click Submit. The stack is complete when it appears in the Stacks list with status of CREATE_COMPLETE.

When the template is successfully uploaded to AWS and the stack creation is complete, the initial discovery scan is started. When the scan is complete, you can view the discovered assets in Asset Inventory.

1.6.6.1.2 | Configure AWS integration instances and monitor integration instance health

Abstract

Enable automations from Data Sources and monitor AWS integration instance health.

You can streamline and simplify configuring AWS integration instances within the Data Sources page. This includes granting the necessary permissions for the platform to execute commands, scripts, and playbooks as part of issue response. All automation permissions are added to the Terraform as part of the setup process.

Configure a new or existing AWS integration instance

> **NOTE:**
>
> If you have not yet onboarded your cloud integration, see Ingest cloud assets.

You can configure a new AWS integration instance or edit an existing AWS integration instance, for example to enable automations.

1. Navigate to Settings → Data Sources.
2. In the AWS integration row:
   - To configure a new AWS integration instance: Click ⋮ and then click Add New Instance or click View Details and from the New Instance drop down select the AWS cloud service provider.
   - To edit an existing AWS integration instance: Click View Details and then click the configuration pencil icon.
3. (Optional) Under Show advanced settings, select Automation and select a log level for the automation integration logs.

4. If the instance is not enabled, in the row for the AWS integration instance, right-click and select Enable. Alternatively, click the more options icon and select Enable.

5. Manually upload the template (Terraform) to the relevant cloud provider.

   An automation integration instance with the same name as the cloud integration instance is automatically created and automation permissions are automatically updated in the system. For more information, see Ingest cloud assets.

Monitor AWS integration instance health

Monitoring AWS integration instance health ensures continuous, reliable operation, facilitating issue response and improving overall security posture.

1. Navigate to Settings → Data Sources.

2. In the AWS integration instance row, click the View Details link and then click a specific Instance Name.

   From the list of health statuses, you can click the following to see automation instance health status:

   - Permissions: Shows any permission issues or missing permissions for the instance.
   - Automation: Indicates any errors during automation instance creation or configuration.

     > **NOTE:**
     >
     > Currently, automation permission errors or missing automation permissions do not affect the Automation health status. You can view any permission errors or missing permissions in the the Permissions health status.

1.6.6.2 | Onboard Google Cloud Platform

Abstract

Follow the GCP onboarding wizard and Cortex creates a custom authentication template to be executed in GCP.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on or the Cortex Cloud Runtime Security add-on.

Follow this wizard to onboard your Google Cloud Platform (GCP) environment. The GCP onboarding wizard is designed to facilitate the seamless setup of GCP data into Cortex XSIAM. The guided experience requires minimal user input; simply define the scope of your GCP accounts and specify the scan mode. For full control of the setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM generates an authentication template to establish trust to GCP and grant permissions to Cortex XSIAM. The template must be executed in GCP to complete the onboarding process. Execution of the template grants the permissions and includes a component that notifies Cortex XSIAM of the execution details and a new cloud instance is created.

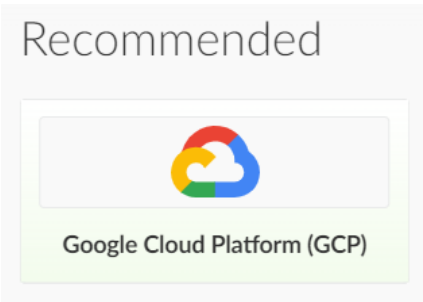
> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - Access to Google Cloud Console
> - Admin user with required admin GCP permissions
> - Enabled the following APIs in the GCP project you are onboarding:
>   - Cloud Resource Manager API
>   - Identity and Access Management (IAM) API
>   - Cloud Pub/Sub API (if audit logs are enabled)

How to onboard GCP:

1. Select Settings → Data Sources.

2. Select Add Data Source.

3. On the Add Data Sources page, search for and select Google Cloud Platform and click Connect.



4. In the onboarding wizard, choose the scope for this data source:

   - Organization: (Default) A collection of GCP projects that are managed centrally.
   - Folder: A GCP folder can contain projects, folders, or a combination of both.
   - Project: A specific GCP project.

5. Choose the Scan Mode:

   - Cloud Scan: (Recommended) Security scanning is performed in the Cortex cloud environment.
   - Scan with Outpost: Security scanning is performed on infrastructure deployed to a cloud account owned by you. If you select this option, choose the outpost account to use for this instance.

     > **NOTE:**
     >
     > Scanning with an outpost may require additional CSP permissions and may incur additional CSP costs.

6. If you selected Organization or Project as the scope, enter its ID.

7. (Optional) Click Show advanced settings to define advanced settings:

   - Instance Name: Enter a unique instance name or leave empty to be automatically populated. The automatic naming convention is the CSP name followed by the ID of the scope unit selected in the onboarding wizard. For example, when onboarding a Google Cloud Platform project, the automatic name would be `GCP-<projectID>` where `<projectID>` is the ID of the project onboarded.
   - Scope Modifications: To allow you to fine tune your GCP data collection, you can modify the scope by including or excluding specific regions. Additionally, if you selected organization or folder as the scope, you can modify the scope by including or excluding specific projects. If you choose to include specific projects, only those specified projects will be included, even if additional projects are added to your GCP environment after onboarding. If you choose to exclude specific projects, any new projects added to your GCP environment after onboarding will be included in the scope. Excluded projects are not visible in Cortex XSIAM.
   - Additional Security Capabilities: Enable additional Cortex security add-ons, if available. This may require additional cloud provider permissions. For detailed information on the permissions required, see Cloud service provider permissions. The additional security capabilities you can enable include:
     - XSIAM analytics: Analyzes your endpoint data to develop a baseline and raise Analytics and Analytics BIOC alerts when anomalies and malicious behaviors are detected.
     - Data security posture management: An agentless multi-cloud data security solution that discovers, classifies, protects, and governs sensitive data.
     - Registry scanning: Scan container registry images for vulnerabilities. malware, and secrets. You can configure your initial preference for scanning your registry. Any newly discovered registry, repository or image in the account will be scanned by default. For more details, see Configure registry scanning
     - Serverless functions scanning (Gen 1 only): Implement serverless scanning to detect and remediate vulnerabilities within serverless functions during the development lifecycle. Seamless integration into CI/CD pipelines enables automated security scans for a continuously secure pre-production environment.
     - Automation: Pre-configures a list of integrations and associated commands to automate security issue responses. Commands can be utilized individually or as part of custom playbooks for issue remediation.
     - Agentless disk scanning: (Recommended) Implement agentless disk scanning to remotely detect and remediate vulnerabilities during the development lifecycle.
   - Log Level: (Optional - for Automation only) Configure the automation integration logging level. Possible values are:
     - Off (Default)
     - Debug
     - Verbose
   - Cloud Tags: Define tags and tag values to be added to any new resource created by Cortex in the cloud environment.
   - Log Collection Configuration: To maximize security coverage, include collection of audit logs (GCP Pub/Sub). This may require additional cloud service provider permissions.For detailed information on the permissions required, see Cloud service provider permissions.

8. Click Save.

9. Download the template file by clicking Download Terraform and then click Close.

   When the file has downloaded, proceed to manually upload the template to GCP.

   > **NOTE:**

> You can see the automatically created automation instance in the Automation & Feed Integrations page under the Cloud Services section, and it will have the same name as the cloud integration instance. The instance is read-only, you can only modify the instance from the Data Sources page.

Required admin GCP permissions

To onboard a GCP organization to Cortex XSIAM, the following permissions are required:

- iam.roles.create
- iam.roles.delete
- iam.roles.get
- iam.roles.list
- iam.roles.update
- iam.serviceAccounts.create
- iam.serviceAccounts.delete
- iam.serviceAccounts.get
- iam.serviceAccounts.getIamPolicy
- iam.serviceAccounts.list
- iam.serviceAccounts.setIamPolicy
- iam.serviceAccounts.update
- logging.sinks.create
- logging.sinks.delete
- logging.sinks.get
- logging.sinks.update
- pubsub.subscriptions.create
- pubsub.subscriptions.delete
- pubsub.subscriptions.getIamPolicy
- pubsub.subscriptions.setIamPolicy
- pubsub.subscriptions.update
- pubsub.topics.create
- pubsub.topics.delete
- pubsub.topics.getIamPolicy
- pubsub.topics.setIamPolicy
- pubsub.topics.update
- resourcemanager.folders.get
- resourcemanager.folders.getIamPolicy
- resourcemanager.folders.list
- resourcemanager.folders.setIamPolicy
- resourcemanager.organizations.get
- resourcemanager.organizations.getIamPolicy
- resourcemanager.organizations.setIamPolicy
- resourcemanager.projects.get
- resourcemanager.projects.getIamPolicy
- resourcemanager.projects.list
- resourcemanager.projects.setIamPolicy

1.6.6.2.1 | Manually upload template to GCP

Abstract

Learn how to manually deploy the Terraform template file in Google Cloud Console.

When you have downloaded the Terraform template file in the onboarding wizard, you must connect to Google Cloud Console to create a stack using the template file.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - A GCP account
> - Permission to create the required resources in Google Cloud Deployment Manager
> - Installed Terraform on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
> - Installed the GCP gcloud CLI tool

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your GCP account using the gcloud CLI:

   ```
   gcloud auth login
   ```

3. Create a directory on your local machine to store and run the Terraform code. If you have more than one GCP connector, you need a separate directory for each one:

   > **NOTE:**
   >
   > The directory you create must be a subdirectory of the home directory.

   ```
   mkdir -p ~/terraform/gcp-connector-1
   ```

4. Navigate to the directory you created and extract the Terraform files. Ensure all necessary Terraform files are present (`main.tf`, `template_params.tfvars`, etc).

   ```
   cd ~/terraform/gcp-connector-1
   tar -xzvf <your_template>.tar.gz
   ```

5. Initialize Terraform in your project directory:

   ```
   terraform init
   ```

6. Apply your Terraform configuration using the downloaded parameter file. When prompted, enter the project ID if you configured one in the onboarding wizard:

   ```
   terraform apply --var-file=template_params.tfvars
   ```

   The Terraform template is deployed.

When the template is successfully uploaded to GCP, the initial discovery scan is started. When the scan is complete, you can view your cloud assets in Asset Inventory.

1.6.6.2.2 | Configure GCP integration instances and monitor integration instance health

Abstract

Enable automations from Data Sources and monitor GCP integration instance health.

You can streamline and simplify configuring GCP integration instances within the Data Sources page. This includes granting the necessary permissions for the platform to execute commands, scripts, and playbooks as part of issue response. All automation permissions are added to the Terraform as part of the setup process.

Configure a new or existing GCP integration instance

> **NOTE:**
>
> If you have not yet onboarded your cloud integration, see Ingest cloud assets.

You can configure a new GCP integration instance or edit an existing GCP integration instance, for example to enable automations.

1. Navigate to Settings → Data Sources.
2. In the GCP integration row:

- To configure a new GCP integration instance: Click ⋮ and then click Add New Instance or click View Details and from the New Instance drop down select the GCP cloud service provider.
- To edit an existing GCP integration instance: Click View Details and then click the configuration pencil icon.

3. (Optional) Under Show advanced settings, select Automation and select a log level for the automation integration logs.
4. If the instance is not enabled, in the row for the GCP integration instance, right-click and select Enable. Alternatively, click the more options icon and select Enable.
5. Manually upload the template (Terraform) to the relevant cloud provider.

   An automation integration instance with the same name as the cloud integration instance is automatically created and automation permissions are automatically updated in the system. For more information, see Ingest cloud assets.

Monitor GCP integration instance health

Monitoring GCP integration instance health ensures continuous, reliable operation, facilitating issue response and improving overall security posture.

1. Navigate to Settings → Data Sources.
2. In the GCP integration instance row, click the View Details link and then click a specific Instance Name.

   From the list of health statuses, you can click the following to see automation instance health status:

   - Permissions: Shows any permission issues or missing permissions for the instance.
   - Automation: Indicates any errors during automation instance creation or configuration.

     > **NOTE:**
     >
     > Currently, automation permission errors or missing automation permissions do not affect the Automation health status. You can view any permission errors or missing permissions in the the Permissions health status.

1.6.6.2.3 | Monitor GCP resources inside service perimeters

Abstract

Learn how to grant authorization to Cortex XSIAM to scan within your GCP service perimeter.

A service perimeter can provide an additional layer of security for your GCP projects. It serves as a fortified boundary around your Google Cloud resources. While resources inside the perimeter can communicate freely, the perimeter is designed to prevent unauthorized communication to Google Cloud services beyond its confines.

To enable Cortex XSIAM to scan assets and resources within your GCP perimeter, you must authorize Cortex XSIAM's identities to access the perimeter from within GCP. If you have a perimeter set up in your GCP project and you have not authorized Cortex XSIAM's identities to scan the perimeter, you will receive the following error:

```
Request is prohibited by organization's policy. vpcServiceControlsUniqueIdentifier: {{<GCP-perimeter-ID>}}
```

> **NOTE:**
>
> Each GCP cloud instance is assigned a scope within GCP. If the scope, whether it be organization, folder, or project, includes any projects with a service perimeter, this procedure must be performed for that cloud instance to authorize Cortex XSIAM to scan the resources in the perimeter.

Obtain Cortex XSIAM identity details

1. In your Cortex XSIAM tenant, select Settings → Data Sources.
2. Hover over the Google Cloud Platform (GCP) row and select View Details.
3. In the Cloud Instances page, identify the GCP instance with the perimeter, right-click it and select Details.
4. In the details pane, click the more options icon and select Authorization Details.
5. The authorization values that you need to add as approved identities in GCP are listed in the Authorization Details dialog box.

Add Cortex XSIAM authorization values to GCP perimeter

1. Log into Google Cloud Platform Console.
2. Navigate to VPC Service Controls.
3. In the list of perimeters, select the perimeter to which you want to grant access to Cortex XSIAM.
4. In the Service perimeter details screen, click Edit.
5. In the Edit service perimeter screen, select Ingress policy.

6. In the Ingress rules pane, click Add an ingress rule.
7. Enter a Title for the ingress rule.
8. In the From section, under Identities, select Select identities & groups.
9. Click Add identities. In the Add identities pane, under Search identities, paste Authorized value #1 from Cortex XSIAM's Authorization Details dialog box. If there are more authorized values, paste each of them under Search identities. Click Add identities.
10. In the To section, under Resources, select Select projects.
11. Click Add projects. In the Add projects pane, select the relevant projects.
12. Under Operations or IAM roles, select All operations.
13. Click Next to add an egress rule.
14. In the Egress rules pane, click Add an egress rule.
15. Enter a Title for the egress rule.
16. In the From section, under Identities, select Select identities & groups.
17. Click Add identities. In the Add identities pane, under Search identities, paste Authorized value #1 from Cortex XSIAM's Authorization Details dialog box. If there are more authorized values, paste each of them under Search identities. Click Add identities.
18. In the To section, under Resources, select Select projects.
19. Click Add projects. In the Add projects pane, select the relevant projects.
20. Click Save. Confirm the changes and click Confirm.

The Cortex XSIAM authorization values have been added as approved identities in GCP.

1.6.6.3 | Onboard Microsoft Azure

Abstract

Follow the Azure onboarding wizard and Cortex creates a custom authentication template to be executed in Azure.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on.

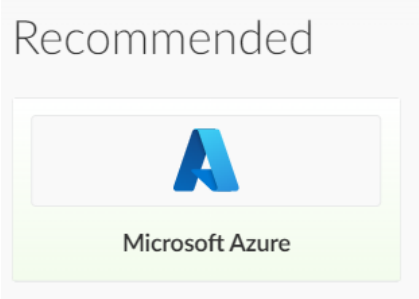
Follow this wizard to onboard your Microsoft Azure environment. The Azure onboarding wizard is designed to facilitate the seamless setup of Azure data into Cortex XSIAM. The guided experience requires minimal user input; simply define the scope of your Azure accounts and specify the scan mode. For full control of the setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM generates an authentication template to establish trust to Azure and grant permissions to Cortex XSIAM. The template must be executed to complete the onboarding process. Execution of the template grants the permissions and includes a component that notifies Cortex XSIAM of the execution details and a new cloud instance is created.

Azure private resources are not currently discoverable.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - An Azure subscription
> - Admin permissions required to onboard Azure
> - Tenant ID and subscription ID. You can view these in Microsoft Azure Portal in Management groups.

How to onboard Azure:

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select Microsoft Azure and click Connect.



4. In the onboarding wizard, choose the scope for this data source. Onboarding of Azure tenants and management groups is performed using a bash script in the Microsoft Azure Resource Manager. Onboarding of Azure subscriptions can be performed using a Terraform script or in Microsoft Azure Resource Manager.
    - Tenant: (Default) A specific instance of Azure Active Directory, which can contain several subscriptions.
    - Management Group: A collection of Azure subscriptions.
    - Subscription: A collection of Azure resources associated with a specific Azure tenant.
5. Choose the Scan Mode:
    - Cloud Scan: (Recommended) Security scanning is performed in the Cortex cloud environment.
    - Scan with Outpost: Security scanning is performed on infrastructure deployed to a cloud account owned by you. If you select this option, choose the outpost account to use for this instance or create a new outpost. For more information on outposts, see Outposts.

        > **NOTE:**
        >
        > Scanning with an outpost may require additional CSP permissions and may incur additional CSP costs.
6. Select an approved tenant ID from the Tenant ID list. If no tenant IDs have been approved, enter the tenant ID. Click Approve in Azure to add Cortex as an approved application on this tenant. When the tenant ID is approved, it appears with a green check next to it.
7. (Optional) Click Show advanced settings to define advanced settings:

- Instance Name: Enter a unique instance name or leave empty to be automatically populated. The automatic naming convention is the CSP name followed by the ID of the scope unit selected in the onboarding wizard. For example, when onboarding an Azure tenant, the automatic name would be `AZURE-<tenantID>` where `<tenantID>` is the ID of the tenant onboarded.
- Scope Modifications: To allow you to fine tune your Azure data collection, you can modify the scope by including or excluding specific regions. Additionally, if you selected tenant or management group as the scope, you can modify the scope by including or excluding specific subscriptions. If you choose to include specific subscriptions, only those specified subscriptions will be included, even if additional subscriptions are added to your Azure environment after onboarding. If you choose to exclude specific subscriptions, any new subscriptions added to your Azure environment after onboarding will be included in the scope. Excluded subscriptions are not visible in Cortex XSIAM.
- Additional Security Capabilities: Enable additional Cortex security add-ons, if available. This may require additional cloud provider permissions. For detailed information on the permissions required, see Cloud service provider permissions. The additional security capabilities you can enable include:
    - XSIAM analytics: Analyzes your endpoint data to develop a baseline and raise Analytics and Analytics BIOC alerts when anomalies and malicious behaviors are detected.
    - Data security posture management: An agentless multi-cloud data security solution that discovers, classifies, protects, and governs sensitive data.
    - Registry scanning: Scan container registry images for vulnerabilities. malware, and secrets. You can configure your initial preference for scanning your registry. Any newly discovered registry, repository or image in the account will be scanned by default. For more details, see Configure registry scanning
    - Serverless functions scanning: Implement serverless scanning to detect and remediate vulnerabilities within serverless functions during the development lifecycle. Seamless integration into CI/CD pipelines enables automated security scans for a continuously secure pre-production environment.
    - Automation: Pre-configures a list of integrations and associated commands to automate security issue responses. Commands can be utilized individually or as part of custom playbooks for issue remediation.
    - Agentless disk scanning: (Recommended) Implement agentless disk scanning to remotely detect and remediate vulnerabilities during the development lifecycle.
- Log Level: (Optional - for Automation only) Configure the automation integration logging level. Possible values are:
    - Off (Default)
    - Debug
    - Verbose
- Cloud Tags: Define tags and tag values to be added to any new resource created by Cortex in the cloud environment.
- Log Collection Configuration: To maximize security coverage, include collection of audit logs (Event Hub). This may require additional cloud service provider permissions. For detailed information on the permissions required, see Cloud service provider permissions.

8. Click Save.
9. To complete the process, download the authentication template:
    - For onboarding Azure tenants and management groups, click Azure Resource Manager to download a tar.gz file and proceed to execute the Azure Resource Manager authentication template.
    - For onboarding Azure subscriptions, click:
        - Download Terraform to download a Terraform file and proceed to execute the Terraform authentication template.
        - Azure Resource Manager to download a JSON file and proceed to deploy the authentication template in Azure Resource Manager.
10. Click Close.

    When the template is successfully executed, the initial discovery scan is started. When the scan is complete, you can view the discovered assets in Asset Inventory.

    > **NOTE:**
    >
    > You can see the automatically created automation instance in the Automation & Feed Integrations page under the Cloud Services section, and it will have the same name as the cloud integration instance. The instance is read-only, you can only modify the instance from the Data Sources page.

Required Azure permissions

To onboard an Azure subscription to Cortex XSIAM, the following permissions are required:

- Microsoft.Resources/deploymentScripts/*
- Microsoft.Resources/subscriptions/resourceGroups/*
- "Microsoft.Authorization/roleAssignments/read
- Microsoft.Authorization/roleAssignments/write
- Microsoft.Authorization/roleAssignments/delete
- Microsoft.Authorization/roleDefinitions/read
- Microsoft.Authorization/roleDefinitions/write
- Microsoft.Authorization/roleDefinitions/delete
- Microsoft.Authorization/roleManagementPolicies/read
- Microsoft.Authorization/roleManagementPolicies/write
- Microsoft.Authorization/roleManagementPolicyAssignments/read
- Microsoft.aadiam/diagnosticsettings/write
- Microsoft.aadiam/diagnosticsettings/read
- Microsoft.aadiam/diagnosticsettings/delete
- Microsoft.aadiam/azureADMetrics/providers/Microsoft.Insights/diagnosticSettings/write
- Microsoft.aadiam/tenants/providers/Microsoft.Insights/diagnosticSettings/write
- Microsoft.Resources/deployments/validate/action
- Microsoft.Insights/DiagnosticSettings/Write
- Microsoft.Resources/deployments/read
- Microsoft.Resources/deployments/write
- Microsoft.Resources/deployments/delete
- Microsoft.Resources/deployments/cancel/action
- Microsoft.Resources/deployments/whatIf/action
- Microsoft.Resources/deployments/operations/read
- Microsoft.Resources/deployments/exportTemplate/action
- Microsoft.Resources/deployments/operationstatuses/read

To onboard an Azure management group or tenant, in addition to the above Azure permissions, the following Azure permissions are required:

- Microsoft.Authorization/elevateAccess/action
- Microsoft.PolicyInsights/remediations/read
- Microsoft.PolicyInsights/remediations/write
- Microsoft.PolicyInsights/remediations/delete
- Microsoft.PolicyInsights/remediations/cancel/action
- Microsoft.PolicyInsights/remediations/listDeployments/read

1.6.6.3.1 | Manually upload template to Microsoft Azure Resource Manager using the CLI

Abstract

Learn how to manually upload the JSON file in Microsoft Azure Resource Manager using the CLI.

When you select the Azure Resource Manager option in the Azure onboarding wizard, you must then execute the Microsoft Azure Resource Manager (ARM) template in ARM using the CLI. This procedure is used for onboarding Microsoft Azure tenants or management groups. For onboarding Microsoft Azure subscriptions, see Manually upload template for Microsoft Azure subscriptions.

> **PREREQUISITE:**
>
> Ensure the Azure CLI tool is installed and you are authorized to create management group policies.

1. Open your local terminal.

2. Create a directory on your local machine to store the tar file. If you have more than one Azure connector, you need a separate directory for each one:

```
mkdir -p ~/azure-connector-1
```

3. Navigate to the directory you created and extract the files.

```
cd ~/azure-connector-1
tar -xzvf <your_template>.tar.gz.
```

4. In a local terminal, run the `main.sh` file:

```
bash main.sh
```

5. When prompted, enter the following values:

   - An existing resource group name that will be used during onboarding
   - The Azure region where you want the resources to be created. (For example, eastus or westus.)
   - The ID of the management group or tenant that you want to onboard
   - The ID of the subscription where the deployment script will run

   When the template is successfully executed, the initial discovery scan is started. When the scan is complete, you can view your cloud assets in Asset Inventory.

1.6.6.3.2 | Manually upload template for Microsoft Azure subscriptions

Abstract

Learn how to manually deploy the Terraform template file in Microsoft Azure or in Microsoft Azure Resource Manager (ARM). Either of these procedures can be used for onboarding Microsoft Azure subscriptions.

You can choose one of the two following methods for executing the authentication template in Microsoft Azure when onboarding Microsoft Azure subscriptions:

Execute the Terraform authentication template

After you have downloaded the Terraform template file in the onboarding wizard, you must log in to Microsoft Azure to execute the template file.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - An Azure subscription.
> - Permission to deploy a custom template and create its resources in Microsoft Azure (Owner or Global Admin).
> - Tenant ID and subscription ID. You can view these in Microsoft Azure Portal in Management groups.
> - Installed Terraform on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
> - Installed the Azure CLI tool.

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your Azure account using the Azure CLI:

```
az login
```

3. Create a directory on your local machine to store and run the Terraform code. If you have more than one Azure connector, you need a separate directory for each one:

```
mkdir -p ~/terraform/azure-connector-1
```

4. Navigate to the directory you created and extract the Terraform files. Ensure all necessary Terraform files are present (`main.tf`, `template_params.tfvars`, etc).

```
cd ~/terraform/azure-connector-1
tar -xzvf <your_template>.tar.gz.
```

5. Initialize Terraform in your project directory:

```
terraform init
```

6. Apply your Terraform configuration using the downloaded parameter file. When prompted, enter the subscription ID:

```
terraform apply --var-file=template_params.tfvars
```

7. When prompted, review the actions the Terraform will perform and approve them by entering `yes`.

The Terraform template is executed.

Deploy the authentication template in Azure Resource Manager

When you select the Azure Resource Manager option in the Azure onboarding wizard, you must then execute the Microsoft Azure Resource Manager (ARM) template in ARM. This procedure is used for onboarding Microsoft Azure subscriptions. For onboarding Microsoft Azure tenants or management groups, see Manually upload template to Microsoft Azure Resource Manager using the CLI.

> **PREREQUISITE:**
>
> Ensure the Azure CLI tool is installed and you are authorized to create management group policies.

1. Open your local terminal.
2. Log in to your Azure account using the Azure CLI:

   ```
   az login
   ```

3. Deploy the template file:

   ```
   az deployment sub create  -g <RESOURCE_GROUP> --location <LOCATION>  --subscription <SUBSCRIPTION_ID> --template-file <JSON_TEMPLATE>
   ```

   where:

   - `<RESOURCE_GROUP>` is an existing resource group name that will be used during onboarding.
   - `<LOCATION>` is the location of the resource group. (For example, eastus or westus.)
   - `<SUBSCRIPTION_ID>` is the ID of the subscription you want to onboard.
   - `<JSON_TEMPLATE>` is the JSON template file that you downloaded at the end of the onboarding wizard.

   The template is deployed.

When the template is successfully executed, the initial discovery scan is started. When the scan is complete, you can view your cloud assets in Asset Inventory.

1.6.6.3.3 | Configure Azure integration instances and monitor integration instance health

Abstract

Enable automations from Data Sources and monitor Azure integration instance health.

You can streamline and simplify configuring Azure integration instances within the Data Sources page. This includes granting the necessary permissions for the platform to execute commands, scripts, and playbooks as part of issue response. All automation permissions are added to the Terraform as part of the setup process.

Configure a new or existing Azure integration instance

> **NOTE:**
>
> If you have not yet onboarded your cloud integration, see Ingest cloud assets.

You can configure a new Azure integration instance or edit an existing Azure integration instance, for example to enable automations.

1. Navigate to Settings → Data Sources.
2. In the Azure integration row:

   - To configure a new Azure integration instance: Click ⋮ and then click Add New Instance or click View Details and from the New Instance drop down select the Azure cloud service provider.
   - To edit an existing Azure integration instance: Click View Details and then click the configuration pencil icon.

3. (Optional) Under Show advanced settings, select Automation and select a log level for the automation integration logs.
4. If the instance is not enabled, in the row for the Azure integration instance, right-click and select Enable. Alternatively, click the more options icon and select Enable.
5. Manually upload the template (Terraform) to the relevant cloud provider.

   An automation integration instance with the same name as the cloud integration instance is automatically created and automation permissions are automatically updated in the system. For more information, see Ingest cloud assets.

Monitor Azure integration instance health

Monitoring Azure integration instance health ensures continuous, reliable operation, facilitating issue response and improving overall security posture.

1. Navigate to Settings → Data Sources.
2. In the Azure integration instance row, click the View Details link and then click a specific Instance Name.

   From the list of health statuses, you can click the following to see automation instance health status:

   - Permissions: Shows any permission issues or missing permissions for the instance.
   - Automation: Indicates any errors during automation instance creation or configuration.

     > **NOTE:**
     >
     > Currently, automation permission errors or missing automation permissions do not affect the Automation health status. You can view any permission errors or missing permissions in the the Permissions health status.

1.6.6.4 | Onboard Oracle Cloud Infrastructure

Abstract

Follow the OCI onboarding wizard and Cortex creates a custom authentication template to be executed in OCI.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on.

Follow this wizard to onboard your Oracle Cloud Infrastructure (OCI) environment. The OCI onboarding wizard is designed to facilitate the seamless setup of OCI data into Cortex XSIAM. This guided experience requires minimal user input. For full control of the setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM generates an authentication template to establish trust to OCI and grant permissions to Cortex XSIAM. Execution of the template completes the onboarding process. The template grants the permissions, includes a component that notifies Cortex XSIAM of the execution details, and a new cloud instance is created.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - Access to Oracle Cloud Infrastructure console
> - Permissions for all of the following are required:
>   - Creation of identity groups (for more information, refer to Managing Groups)
>   - Policies (for more information, refer to How Policies Work)
>   - Tag namespaces in the root compartment (for more information, refer to Tags and Tag Namespace Concepts)

How to onboard OCI:

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select Oracle Cloud Infrastructure (OCI) and click Connect. It might take about 15 minutes until the wizard moves to the next phase.

   The scope for this data source type is always set to Tenancy.
4. When the scanning infrastructure has been created, click Close.
5. Return to the Data Sources page, and click the Oracle Cloud Infrastructure instance that you are adding.
6. Optionally, enter a unique instance name.

   If you don't enter a name, the wizard will apply the default name, `OCI-<TENANCY_OCID>`.
7. (Optional) To define advanced settings, click Show advanced settings and configure as needed:

   - Scope Modifications: You can modify the scope by including or excluding specific Compartments. If you choose to include specific compartments, only the specified compartments and their sub-compartments will be included. This setting will affect future sub-compartments added to your OCI environment after onboarding. If you choose to exclude specific compartments, this setting will also affect their sub-compartments.

     Note:

     The root compartment is always onboarded, and only sub-compartment scope can be modified.

     Excluded compartments are not visible in Cortex XSIAM.
8. Click Save.
9. Download the OCI authentication template by clicking Download Terraform.

10. Click Close, and then follow the instructions to manually upload the template to OCI.

> **NOTE:**
>
> You can see the automatically created automation instance in the Automation & Feed Integrations page under the Cloud Services section, and it will have the same name as the cloud integration instance. The instance is read-only; you can only modify the instance from the Data Sources page.

1.6.6.4.1 | Manually upload template to OCI

Abstract

Learn how to manually deploy the Terraform template files in Oracle Cloud Infrastructure (OCI).

When you have downloaded the Terraform template files in the onboarding wizard, you must log in to the Oracle Cloud Infrastructure (OCI) CLI tool to deploy the template file. For more information about the OCI CLI tool, refer Oracle documentation.

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - An Oracle Cloud Infrastructure account and the tenancy OCID.
> - Permission to deploy a custom template and create its resources in OCI.
> - Installed Terraform on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
> - Installed the OCI CLI tool, and authenticated with a key pair or token-based credentials.

1. Open your local terminal (command prompt).
2. Create a directory on your local machine to store and run the Terraform code. If you have more than one OCI connector, you need a separate directory for each one. For example:

   ```
   mkdir -p ~/terraform/oci-connector-1
   ```

3. Navigate to the directory you created and extract the Terraform files. Ensure all necessary Terraform files are present (`main.tf`, `template_params.tfvars`, and so on). For example:

   ```
   cd ~/terraform/oci-connector-1
   tar -xzvf <your_template>.tar.gz.
   ```

4. Initialize Terraform in your project directory:

   ```
   terraform init
   ```

   It might take several seconds until the initialization is complete.

5. Log in to OCI:

   ```
   oci session authenticate
   ```

   An Oracle Cloud window is displayed.

6. Click your account.
7. Log in to your account, then close the window and return to your local terminal.
8. Apply your Terraform configuration using the downloaded parameter file. When prompted to enter a value, enter the tenancy OCID.

   ```
   terraform apply --var-file=template_params.tfvars
   ```

9. When prompted, review the actions the Terraform will perform, and approve them by entering `yes`.

   The Terraform template is deployed.

When the template is successfully uploaded to OCI, the initial discovery scan starts. When the scan is complete, you can view your cloud assets in Asset Inventory. You can also view details about the instance by hovering over the instance on the Data Sources page, and then clicking View Details.

1.6.6.5 | Manually connect a cloud instance

When onboarding your cloud instance using the onboarding wizard, after you download the authentication template and execute it in your cloud environment, notification is sent to Cortex XSIAM and a cloud instance is created. This connection between your cloud environment and the Cortex XSIAM cloud instance typically occurs automatically.

There are several scenarios when the instance should be connected manually:

- You executed the template in your cloud environment and your environment is an air-gapped network. In this case, the notification to create the instance in Cortex XSIAM does not happen.
- You have executed the template, but the instance has not appeared in Cloud Instances. This is often due to connectivity or firewall issues.
- You have a specific need to connect the instance manually.

To manually connect a cloud instance, you need to identify the pending instance you want to connect. In Cloud Instances, remove the default filter that excludes pending instances. Right-click on a pending instance and select View Details to see the configuration details of that specific pending instance. After you have identified the pending instance you want to connect manually, right-click and select Manually connect an instance. For more information on pending instances, see Pending cloud instances.

AWS

In AWS Management Console, navigate to CloudFormation. Use the following table to guide you on where to obtain the necessary input for the manual onboarding. Not every field appears in every manual onboarding instance.

Connect Instance Input Field	Value
Organization ID	Onboarded organization ID.
Organizational Unit ID	Onboarded organizational unit ID.
Account ID	Onboarded account ID.
Role ARN	The value of Outputs → CORTEXXDRARN.
External ID	The value of Parameters → ExternalID.
Audit Logs SQS URL	The value of Resources → CloudTrailLogsQueue.
Audit Logs Role ARN	The value of Resources → CloudTrailReadRole → ARN.
Audit Logs Audience	Automatically populated.
Outpost Scanner Role ARN	The value of Resources → CortexPlatformScannerRole → ARN.

GCP

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your GCP account using the gcloud CLI:

   ```
   gcloud auth login
   ```

3. Display the values of all defined output variables in your Terraform configuration, formatted as a JSON object:

   ```
   terraform output -json
   ```

Use the following table to guide you on which values in the output map to the necessary input for the manual onboarding. Not every field appears in every manual onboarding instance.

Connect Instance Input Field	Value
Organization ID	organization_id.value

Connect Instance Input Field	Value
Project ID	project_id.value
Folder ID	folder_id.value
Service Account Email	service_account_email.value
Audit Logs Audit Pubsub Subscription ID	resources_data.value.AUDIT_LOGS.audit_pubsub_subscription_id
Audit Logs Service Account Email	resources_data.value.AUDIT_LOGS.audit_service_account_email
Outpost Scanner Service Account Email	resources_data.value.OUTPOST_SCANNER.outpost_scanner_service_account_email

Azure with Terraform

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your Azure account using the Azure CLI:

   ```
   az login
   ```

3. Display the values of all defined output variables in your Terraform configuration, formatted as a JSON object:

   ```
   terraform output -json
   ```

Use the following table to guide you on which values in the output map to the necessary input for the manual onboarding. Not every field appears in every manual onboarding instance.

Connect Instance Input Field	Value
Resource Group Location (only for subscription scope)	Onboarded resource group location
Resource Group Name	Automatically populated
Audit Logs Audience	Automatically populated
Audit Logs Storage Account Name	resources_data.value.AUDIT_LOGS.storage_account_name
Audit Logs Tenant ID	Automatically populated
Audit Logs Client ID	resources_data.value.AUDIT_LOGS.client_id
Audit Logs Namespace	resources_data.value.AUDIT_LOGS.namespace
Audit Logs Eventhub Name	resources_data.value.AUDIT_LOGS.eventhub_name
Audit Logs Azure Audit Eventhub Consumer Group Name	resources_data.value.AUDIT_LOGS.azure_audit_eventhub_consumer_group_name

Azure Portal

- Navigate to the Microsoft Azure Portal and log in.

Use the following table to guide you on which values in the output map to the necessary input for the manual onboarding. Not every field appears in every manual onboarding instance.

Connect Instance Input Field	Value
Resource Group Location (only for subscription scope)	Onboarded resource group location
Resource Group Name	Automatically populated
Audit Logs Audience	Automatically populated
Audit Logs Storage Account Name	Navigate to Storage accounts and filter by resource group.
Audit Logs Tenant ID	Automatically populated
Audit Logs Client ID	Navigate to App registrations and sort by time. The default name starts with "auditlogsapp".
Audit Logs Namespace	Navigate to Event Hubs and filter by resource group.
Audit Logs Eventhub Name	Navigate to Event Hubs and select the Event Hub Namespace. Under Event Hubs, take the value in the Name column.
Audit Logs Azure Audit Eventhub Consumer Group Name	Navigate to Event Hubs -and select the Event Hub Namespace and then the Event Hub. Under Consumer Groups, use the value in the Name column, but not '$Default'.

1.6.6.6 | Outposts

Abstract

An outpost enables you to have security scans performed on infrastructure in a cloud account owned by you.

Cortex XSIAM allows you to have security scans performed in an outpost, on infrastructure deployed to a cloud account owned by you. The outpost is a set of dedicated cloud services for Cortex XSIAM for the purpose of scanning your workloads while respecting your data security and residence requirements.

The cloud account should be a dedicated account for the outpost, free from other resources. Each cloud account can host only one outpost. Using an outpost requires additional cloud provider permissions and may incur additional cloud costs.

To use an outpost for your cloud workload scanning, under Scan Mode in the CSP onboarding wizard, select Scan with Outpost.

To view all outposts and their details, navigate to Settings → Data Collection → Outposts. In the Outposts page, you can edit and delete outposts. You can also view the outpost instance.

Create an outpost

Follow this wizard to create a CSP outpost. Cortex XSIAM creates an authentication template to establish trust to the CSP and grant the necessary permissions to Cortex XSIAM. The template must be executed in the CSP to complete the outpost creation process.

You can create a new outpost by navigating to Settings → Data Collection → Outposts and clicking New Outpost. Alternatively, you can create a new outpost in the cloud onboarding wizard after choosing the Scan with Outpost option. Under Choose Outpost, click Select outpost account, and then click Create a new outpost.

> **NOTE:**

- We recommend you use a dedicated account for the outpost, free from other resources.
- You can only onboard one outpost for each account.

AWS

1. (Optional) Define tags and tag values to be added to any new resource created by Cortex in the cloud environment. Click Next.
2. Click Download Terraform to download the Terraform template file.

   Execute the Terraform template in the CSP to create the outpost.

GCP

1. Enter the project ID of the GCP project.
2. (Optional) Define tags and tag values to be added to any new resource created by Cortex in the cloud environment. Click Next.
3. Click Download Terraform to download the Terraform template file.

   Execute the Terraform template in the CSP to create the outpost.

Azure

NOTE:

When creating an outpost for a specific Azure subscription, the outpost account must be in the same Azure organization as the monitored subscriptions.

1. Enter the tenant ID of the Azure tenant in which you want to establish the outpost.
2. (Optional) Define tags and tag values to be added to any new resource created by Cortex in the cloud environment. Click Next.
3. Click Download Terraform to download the Terraform template file.

   Execute the Terraform template in the CSP to create the outpost.

Execute the template in the CSP to finalize the outpost

When you have downloaded the Terraform template file in the onboarding wizard, log in to the CSP and execute the template file.

AWS

PREREQUISITE:

Before you begin, ensure you have:

- An AWS account
- Permission to create a stack and its resources in AWS
- Installed Terraform on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
- Installed the AWS CLI tool and configured your profile with the `aws configure sso` wizard.

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your AWS account using the AWS CLI:

   ```
   aws sso login --profile <my-profile>
   ```

   Where `<my-profile>` is the profile you configured with the `aws configure sso` wizard.
3. Create a directory on your local machine to store and run the Terraform code. If you are creating more than one outpost, you need a separate directory for each one:

   ```
   mkdir -p ~/terraform/aws-outpost-1
   ```
4. Navigate to the directory you created and extract the Terraform files.

   ```
   cd ~/terraform/aws-outpost-1
   tar -xzvf <your_template>.tar.gz
   ```
5. Initialize Terraform in your project directory:

   ```
   terraform init
   ```
6. Apply your Terraform configuration using the downloaded parameter file. When prompted, enter the subscription ID:

   ```
   terraform apply --var-file=template_params.tfvars
   ```
7. When prompted, review the actions the Terraform will perform and approve them by entering `yes`.

The Terraform template is deployed and your outpost is created. To view all outposts and their details, navigate to Settings → Data Collection → Outposts.

GCP

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - A GCP account
> - Permission to create the required resources in Google Cloud Deployment Manager
> - Installed Terraform on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
> - Installed the GCP gcloud CLI tool

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your GCP account using the gcloud CLI:

   ```
   gcloud auth login
   ```

3. Create a directory on your local machine to store and run the Terraform code. If you are creating more than one outpost, you need a separate directory for each one:

   ```
   mkdir -p ~/terraform/gcp-outpost-1
   ```

4. Navigate to the directory you created and extract the Terraform files.

   ```
   cd ~/terraform/gcp-outpost-1
   tar -xzvf <your_template>.tar.gz
   ```

5. Initialize Terraform in your project directory:

   ```
   terraform init
   ```

6. Apply your Terraform configuration using the downloaded parameter file. When prompted, enter the project ID:

   ```
   terraform apply --var-file=template_params.tfvars
   ```

7. When prompted, review the actions the Terraform will perform and approve them by entering `yes`.

   The Terraform template is deployed and your outpost is created. To view all outposts and their details, navigate to Settings → Data Collection → Outposts.

Azure

> **PREREQUISITE:**
>
> Before you begin, ensure you have:
>
> - An active Azure subscription.
> - Installed the Azure CLI tool.
> - Permission to deploy a custom template and create its resources in Microsoft Azure ("Owner" or "Contributor" on the designated outpost subscription scope, and Active Directory "Cloud Application Administrator" or "Application Administrator" privileged roles).
> - Installed Terraform 1.9.4 or above on your local machine. You can download Terraform from the official Terraform website and follow the installation instructions for your operating system.
> - A static egress IP assigned to the machine running this Terraform. This is used to configure the Azure Storage IP whitelist (Recommended). Without this, future runs of this Terraform may fail on Azure storage configurations.

1. Open your local terminal (Command prompt, PowerShell, or Terminal).
2. Log in to your Azure account using the Azure CLI:

   ```
   az login
   ```

3. If prompted, select the subscription_id of the designated subscription, or run:

   ```
   az account set --subscription <subscription_id>
   ```

   Where `<subscription_id>` is the subscription ID of the designated subscription.

4. Create a directory on your local machine to store and run the Terraform code. If you are creating more than one outpost, you need a separate directory for each one:

   ```
   mkdir -p ~/terraform/azure-outpost-1
   ```

5. Navigate to the directory you created and extract the Terraform files.

```
cd ~/terraform/azure-outpost-1
tar -xzvf <your_template>.tar.gz
```

6. Initialize Terraform in your project directory:

```
terraform init
```

7. Apply your Terraform configuration using the downloaded parameter file. When prompted, enter the subscription ID:

```
terraform apply --var-file=template_params.tfvars
```

8. When prompted for `var.storaage_account_ip_whitelist`, you can leave it empty to enable access from any public IP to the storage accounts. We recommend you to limit access to selected IPs. To limit access, enter a comma-separated list of public IP addresses, including your local machine's egress IP (to enable the completion of the Terraform run). For example: `8.8.8.8, 8.8.4.4`

9. Review the actions the Terraform will perform and approve them by entering `yes`.

10. It is important to create a backup of the Terraform state file using one of the following methods:

    Backup the `terraform.tfstate` and `terraform.tfstate.backup` files or use Terraform backend to save the state.

    - Create copies of the `terraform.tfstate` and `terraform.tfstate.backup` files. These can then be moved to the working folder to allow Terraform to upgrade or destroy the created resources as necessary.
    - Ensure you're using a backend block in your Terraform configuration. For more information, see Backend block configuration overview.

    The Terraform template is deployed and your outpost is created. To view all outposts and their details, navigate to Settings → Data Collection → Outposts.

After creating the outpost

Once you have executed the template in your cloud service provider, the necessary permissions are granted and a notification is sent to Cortex XSIAM with the execution details. A new outpost is created in pending status and can be viewed in the Outpost page at Settings → Data Collection → Outposts.

Troubleshooting

If you have successfully executed the template in your cloud service provider and no new outpost has been created, verify that your internet connection is active. An active internet connection is necessary for the notification to be sent to Cortex XSIAM to create the new outpost. If you are unable to establish an internet connection, contact customer support for a manual workaround.

1.6.6.7 | Container Registry Scanning

1.6.6.7.1 | Overview of container registry scanning

Registry scanning identifies vulnerabilities, malware, and secrets, ensuring comprehensive protection for containerized applications across various cloud environments without requiring manual intervention.

Container Registry Scanning automatically detects and scans container registries within your onboarded cloud accounts, including Amazon Elastic Container Registry (ECR), Azure Container Registry (ACR), and Google Artifact Registry (GAR). Also, Runtime Security supports registry scanning for JFrog Artifactory and Docker V2 registries where you can manually onboard them as new data connectors.

After you onboard your container registries, Runtime Security ensures that all containers and images are scanned at regular intervals and that you are notified about any deviation from your security policies and best practices.

1.6.6.7.1.1 | Registry Components

To understand how container registry scanning works, it's essential to understand its core components:

- Container registry: A container registry is a service for publishing, maintaining, and securely distributing container images, providing a centralized hub for managing and accessing containerized application components across your organization. This scanning helps to enable proactive identification and remediation of security risks before deployment which means you will be using only trusted and compliant images in production environments.
- Container image repository: Within a container registry, container images are organized into multiple repositories to improve management, access control, collaboration, and security isolation. Each repository should ideally contain images related to a specific application, service, or project, allowing for granular permissioning and security policies. Images within a repository often share a common base image or purpose, making it easier to apply consistent security controls across related components.
- Image Tags: Image tags are essential for identifying and managing container image versions within a repository, enabling the selection and deployment of appropriate builds. From a security perspective, tags facilitate tracking vulnerable images, deploying patched versions, and maintaining image provenance for auditing. There are two common formats for referencing image tags:
    - image:tag – A human-readable label that can be reassigned to different versions. For example, myapp:latest or myapp:v1.0.0.
    - image@sha – A cryptographic hash that provides an immutable reference to a specific image version. For example, myapp@sha256:abc123.

While human-readable tags like myapp:latest (reassignable) and myapp:v1.0.0 are common, using immutable tags such as myapp@sha256:abc123 provides a cryptographically secure and verifiable reference, crucial for ensuring the integrity and trustworthiness of deployed images.

- Image Digest: A cryptographic digest (SHA-256 hash) uniquely identifies a container image's content, providing a strong guarantee of immutability. Unlike user-defined image tags, which can be reassigned, using the digest as a tag ensures that even if an image is renamed or retagged, its content remains verifiably identical, making it a critical element for security auditing and ensuring the integrity of deployed applications. Relying on image digests helps prevent potential supply chain attacks where malicious actors might attempt to replace images with compromised versions.

1.6.6.7.1.2 | How Container Registry Scanning Works

The process of container registry scanning consists of three key phases: discovery, scanning, and evaluation.

1. Discovery involves detecting registries, repositories, and image tags within an environment. This step ensures that all container images, regardless of their source, are accounted for and available for analysis.
2. Scanning detects vulnerabilities, malware, and secrets in the container images.
3. Evaluation creates compliance findings based on the scan results. These findings identify vulnerabilities, compliance violations, or potential threats that require remediation before an image is deployed. Scan results are evaluated for vulnerabilities, malware and secrets, creating findings accordingly

1.6.6.7.2 | Configure registry scanning

You can enable and configure container registry for managed registries, such as Amazon Elastic Container Registry (ECR), Azure Container Registry (ACR), and Google Artifact Registry (GAR) when you onboard the respective cloud accounts.

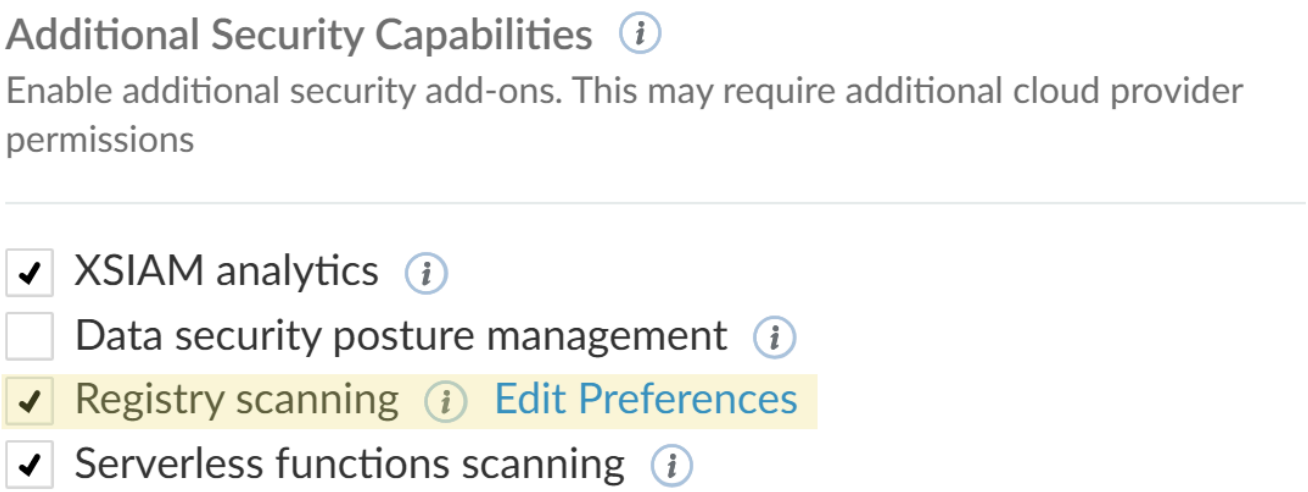
You can also modify an already onboarded account to enable and configure registry scanning for that account as an additional security capability to scan images for vulnerabilities, malware, and secrets.

Configuring registry scanning ensures that only verified and compliant images are deployed across your cloud environments.

> **PREREQUISITE:**
>
> Ensure that you have performed the all steps till Additional Security Capabilities as listed in the onboarding wizard for the required CSP:
>
> - Onboard Amazon Web Services
> - Onboard Google Cloud Platform
> - Onboard Microsoft Azure

To configure registry scanning, do the following:

1. Under Additional Security Capabilities, select Registry Scanning, then click Edit Preferences.



2. In Initial Scan Configuration, set your scanning process to focus on recently added or modified container images and exclude older ones that do not align with your current scanning objectives. This setting helps avoid unnecessary scans. Choose one of the following options:
   - All: Scans all container images, including all versions (tags), in all discovered repositories.
   - Latest Tags: Scans only images tagged 'latest' in all discovered repositories.
   - Days Modified: Scans container images created or modified in the last few days. You can select a range of up to 90 days for the scan.

3. Select Save.

   After configuring your container registers, a new discovery scan is started. When the scan is complete, you can view the scanned images in Container Image page. For more details, see Container Image assets.

1.6.6.7.3 | Modify the container registry scanning scope

Using the Modify Scanning Scope option, you can define conditions to automatically exclude selected scopes from scanning. These conditions can be based on the registry, repository, or tag. After you set the scope, the exclusion conditions are automatically applied to newly discovered images in the account.

To modify the scanning scope, do the following:

1. Navigate to Settings → Data Sources.
2. In the Cloud Provider section, locate the provider where your assets are stored and click View Details.
3. On the Cloud Instances page, click the instance name for which you want to modify the scope.
4. Under the Accounts section, select the account, right-click, and choose Edit.
5. In the Registry Scanning Scope section, click Modify Scanning Scope.
6. From the list of images, select the image you want to modify.
7. Alternatively, you can also filter for a specific image by clicking the Filter icon and selecting Registry, Repository ,or Tags option and then adding the desired value to refine your search.

   The search results are applied automatically, even if you do not select Save.
8. Click Save to confirm your modifications.

This ensures that the specified scanning scope is customized based on your needs.

1.6.6.7.4 | Scan re-evaluation process

After the initial scan has been completed, the scan re-evaluation process ensures that container images remain secure over time without requiring a full re-scan.

Instead of manually triggering new scans, the scan re-evaluation process process automatically reassesses existing scan results every 24 hours using the latest threat intelligence feeds. This approach reduces the need for resource-intensive re-scans, while maintaining up-to-date security assessments.

By continuously monitoring container images for emerging threats, organizations can proactively mitigate risks and ensure compliance with security best practices.

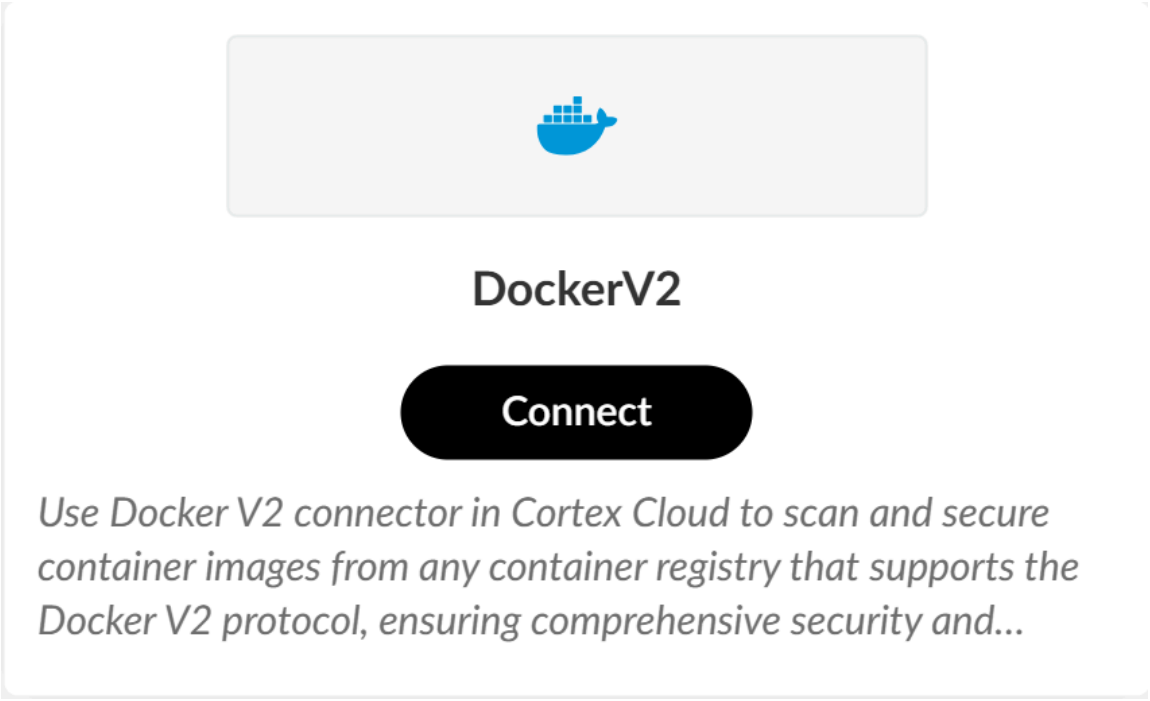
1.6.6.7.5 | Connect Docker V2 compliant container registry

A Docker V2-compliant registry is a registry service that complies with the specifications and requirements outlined in the Docker Registry HTTP API V2. This API defines the protocol for interacting with a Docker registry, a repository where Docker images are stored and from which they can be pulled or pushed.

Follow the wizard to use the Docker V2 connector in Cortex Cloud to scan and secure container images from any container registry that supports the Docker V2 protocol, ensuring comprehensive security.

How to connect Docker V2

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select Docker V2, and then select Connect.



4. The Instance Name is automatically populated. You can change it to a more meaningful name.
5. Choose the Scan Mode, and then follow the steps for that mode to configure the connection.

   Cloud Scan

   Security scanning is done in the Cortex cloud environment when you select this mode.

   1. Select a Cloud Provider to initialize registry scanning.
   2. Select the Region where the registry is hosted.
   3. (Optional) Enable Allow access by IP's to specify a static IP address for the scanner to use. Make sure the static IP is allowed through your firewall so the scanner can access the registry during the scanning process.
   4. Enter the Registry URL. This must match the URL you use with the docker login command.

      For example: `docker login registry-1.docker.io`

      Equivalent URL: `https://registry-1.docker.io/`
   5. Under Authentication Method, enter the Username and Password of the registry that you want to connect.
   6. Select Next.

   Scan with Outpost

   Security scanning is done on infrastructure deployed to a cloud account that you own. This mode requires additional cloud provider permissions and may incur extra costs.

   > **PREREQUISITE:**
   >
   > Ensure an Outpost is connected to your tenant. Outposts

   1. Choose a Cloud Provider to initialize registry scanning.

      > **NOTE:**
      >
      > If you choose Azure as the Cloud Provider, you must also select the Tenant Id. The Tenant Id is required to approve Cortex as an enterprise application in your Azure tenant.
   2. Choose Outpost account to use for this instance. If no Outposts are shown, you can Create a new one. For more details, see Outposts.

      > **NOTE:**
      >
      > If you choose Azure as the cloud provider, only Outposts associated with the selected tenant ID are displayed.
   3. Select the Region where the registry is hosted.
   4. (Optional) Enable Allow access by IP's if you want to specify a static IP address for the scanner to use. Make sure the static IP is allowed through your firewall so that the scanner can access the registry during the scanning process.

5. Enter the Registry URL. This must match the URL you use with the docker login command.

   For example: `docker login registry-1.docker.io`

   Equivalent URL: `https://registry-1.docker.io/`

6. Under Authentication Method, enter the Username and Password of the registry that you want to connect.

7. Select Next.

Scan with Broker VM

Security scanning in private networks is done using broker VM infrastructure when you select this mode.

> **NOTE:**
>
> FedRAMP is not supported for this mode.

> **PREREQUISITE:**
>
> Ensure one of the following is configured:
>
> - Set up and configure Broker VM.
> - Configure High Availability Cluster.

1. Choose a Scan with Broker VM mode to initiate registry scanning. You can select either a standalone Broker VM or a High Availability (HA) Cluster.

2. Select Applicable Broker VMs.

   Choose the appropriate Broker VM or Cluster from the list configured in your tenant.

   > **NOTE:**
   >
   > - The list of Broker VMs displays only VMs that support registry scanning.
   > - The list of high-availability Clusters displays only clusters that contain at least one VM supporting registry scanning.
   > - The registry scanning status for each VM appears in brackets if it was previously activated for that specific VM.

   If the list does not display any Broker VMs or clusters, Add New Broker VM or Add New Cluster. For more details, see Set up and configure Broker VM.

3. Enter the Registry URL. This must match the URL you use with the docker login command.

   For example: `docker login registry-1.docker.io`

   Equivalent URL: `https://registry-1.docker.io/`

4. Under Authentication Method, enter the Username and Password of the registry that you want to connect.

5. Select Next.

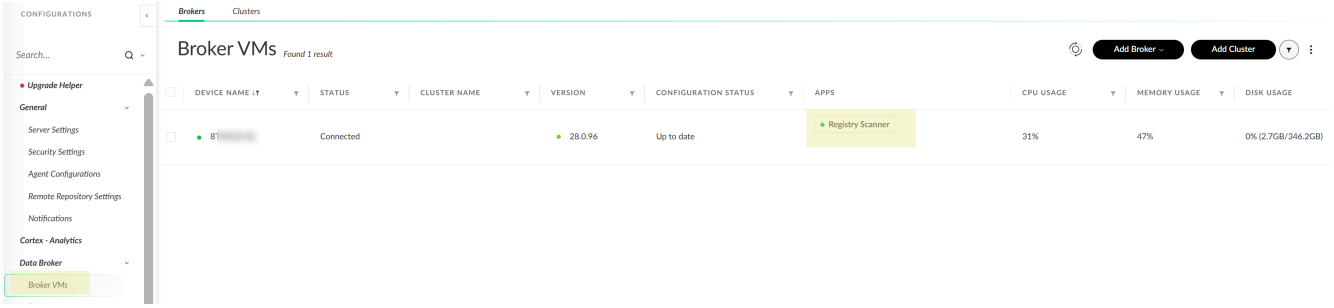
6. In the Initial Scan Configuration, set your scanning process to focus on recently added or modified container images and exclude older ones that do not align with your current scanning objectives. This setting helps avoid unnecessary scans. Choose one of the following options:

   - All: Scans all container images, including all versions (tags), in all discovered repositories.
   - Latest Tag: Scans only images tagged 'latest' in all discovered repositories.
   - Days Modified: Scans container images that have been created in the last few days. You can select a range of up to 90 days for the scan.

7. Select Save.

   When the Docker V2 data source is saved successfully, a new data connector is created, and the initial discovery scan begins. The connection process may take up to 15 minutes. After the scan is complete, you can view the scanned details on the Container Images Inventory page. For more details, see Container Images assets.

   If you have selected the Scan with Broker VM option, then a Registry Scanner applet is created on the selected Broker VM or Cluster. For details, see Verify Registry Scanner connection.



After successfully adding a connector, go to the Docker V2 Instances page to check the connector status and repository scan details.

You can also modify the connector settings and configure the scanning scope for images in the connected registry on this page.

To manage the connector, follow these steps:

1. Select Settings → Data Sources.
2. Find the Docker V2 instance from the list of 3rd Party Data Sources connectors, or use Search.
3. In the Docker V2 instance row, select View Details. The Docker V2 Instances page appears.
4. On the Docker V2 Instances page, you can filter the results by any heading and value. You can also create a new instance by selecting + Add Instance and following the onboarding wizard to define the settings.
5. Select an instance name to open the details pane. The details pane contains the following granular information:

Instance Details	Description
Status	Shows the status of the connector: Connected, Error, Warning, Disabled, or Pending.
Applet Status on Broker VM	Shows the status of the Registry Scanner applet on the Broker VM page. This status is visible only when the Scan with Broker VM mode is selected.
Repositories	Shows the number of scanned repositories in the registry.
Scan Mode	Shows the selected scan mode for the data connector, such as Cloud Scan, Scan with Outpost, or Scan with Broker VM.
Security Capabilities	Shows a breakdown of the security capabilities enabled on the instance and their individual statuses. For example, select Registry Scanning when it shows a warning or error status to see the open errors and issues that contributed to the status.

6. You can also perform actions on each Docker V2 instance: For example, select the  (pencil) icon to Edit the instance, or select the  (three dots) icon to Add image scope, Delete, or Disable the instance as follows:

Action	Instructions
Edit	Edit the Docker V2 instance. **NOTE:** • If you selected Scan with Broker VM mode, you can't change to a different scan mode (such as Cloud Discovery or Scan with Outpost) when you edit the instance. • When editing an instance configured for Scan with Broker VM, you must re-enter your authentication credentials, including Username, Password, and CA certificate.

Action	Instructions
Add image scope	Define conditions to automatically include specific images in scanning. Conditions can be based on Repository or Tags. These conditions apply automatically to newly discovered images in the account.
Delete	Removes the connector.
Disable	Stops image scanning for the connector without deleting it.

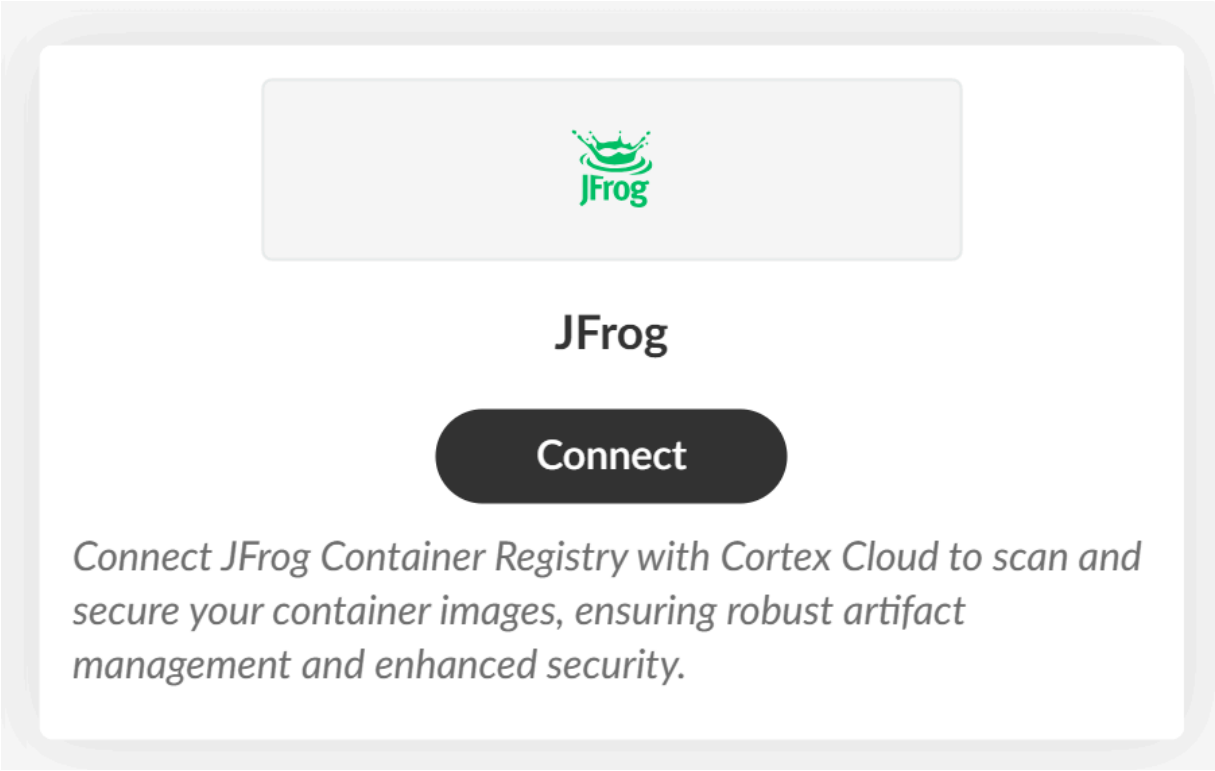
1.6.6.7.6 | Connect JFrog container registry

Follow the wizard to connect your JFrog Container Registry with Cortex Cloud.

Cortex Cloud allows you to scan and secure your container images from vulnerabilities, malware, and secrets after you authenticate and connect your JFrog account. This process ensures robust artifact management and enhanced security.

How to connect JFrog

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select JFrog, and then select Connect.



4. The Instance Name is automatically populated. You can change it to a more meaningful name.
5. Choose the Scan Mode, and then follow the steps provided for that mode to configure the connection.

   Cloud Scan

   Security scanning is done in the Cortex cloud environment when you select this mode.

   1. Select a Cloud Provider to initialize registry scanning.
   2. Select the Region where the registry is hosted.
   3. (Optional) Enable Allow access by IPs to specify a static IP address for the scanner to use. Make sure the static IP is allowed through your firewall so the scanner can access the registry during the scanning process.

4. Choose the relevant Account Type for JFrog deployments:

   JFrog Cloud (Saas)

   a. Enter your JFrog Account Name.

      For example, the scanner connects to `https://myaccount.jfrog.io`, where `<myaccount>` is your actual account name.

   b. Under Authentication Method, enter your JFrog account credentials (Username and Password) for authentication.

   JFrog Self-Hosted

   a. Enter the JFrog Artifactory URL as the Registry URL.

      For example, `https://artifactory.example.com/artifactory`, where `<artifactory.example.com>` is your server's domain or IP address.

   b. Under Authentication Method, enter your JFrog user credentials (Username and Password) for authentication.

   c. (Optional) Expand Show Advanced Settings, and then enter the CA certificate if your JFrog Artifactory server uses a custom CA to sign the server certificate.

5. Select Next.

Scan with Outpost

Security scanning is done on infrastructure deployed to a cloud account that you own. This mode requires additional cloud provider permissions and may incur extra costs.

> **PREREQUISITE:**
>
> Ensure an Outpost is connected to your tenant.

1. Choose a Cloud Provider to initialize registry scanning.

   > **NOTE:**
   >
   > If you choose Azure as the Cloud Provider, you must also select the Tenant Id. The Tenant Id is required to approve Cortex as an enterprise application in your Azure tenant.

2. Choose Outpost account to use for this instance. If no Outposts are shown, you can Create a new one. For more details, see Outposts.

   > **NOTE:**
   >
   > If you choose Azure as the cloud provider, only Outposts associated with the selected tenant ID are displayed.

3. Select the Region where the registry is hosted.

4. (Optional) Enable Allow access by IPs if you want to specify a static IP address for the scanner to use. Make sure the static IP is allowed through your firewall so that the scanner can access the registry during the scanning process.

5. Choose the relevant Account Type for JFrog deployments:

   JFrog Cloud (Saas)

   a. Enter your JFrog Account Name.

      For example, the scanner connects to `https://myaccount.jfrog.io`, where `<myaccount>` is your actual account name.

   b. Under Authentication Method, enter your JFrog account credentials (Username and Password) for authentication.

   JFrog Self-Hosted

   a. Enter the JFrog Artifactory URL as the Registry URL.

      For example, `https://artifactory.example.com/artifactory`, where `<artifactory.example.com>` is your server's domain or IP address.

   b. Under Authentication Method, enter your JFrog user credentials (Username and Password) for authentication.

   c. (Optional) Expand Show Advanced Settings, and then enter the CA certificate if your JFrog Artifactory server uses a custom CA to sign the server certificate.

6. Select Next.

Scan with Broker VM

Security scanning in private networks is done using broker VM infrastructure when you select this mode.

> **PREREQUISITE:**

- Set up and configure Broker VM
- Configure High Availability Cluster

1. Choose a Scan with Broker VM mode to initiate registry scanning. You can select either a standalone Broker VM or a High Availability (HA) Cluster.
2. Select Applicable Broker VMs.

   Choose the appropriate Broker VM or Cluster from the list configured in your tenant.

   NOTE:

   - The list of Broker VMs displays only VMs that support registry scanning.
   - The list of high-availability Clusters displays only clusters that contain at least one VM supporting registry scanning.
   - The registry scanning status for each VM appears in brackets if it was previously activated for that specific VM.

   If the list does not display any Broker VMs or Clusters, Add New Broker VM or Add New Cluster. For more details, see Set up and configure Broker VM.
3. Choose the relevant Account Type for JFrog deployments:

   JFrog Cloud (Saas)

   a. Enter your JFrog Account Name.

   For example, the scanner connects to `https://myaccount.jfrog.io`, where `<myaccount>` is your actual account name.

   b. Under Authentication Method, enter your JFrog account credentials (Username and Password) for authentication.

   JFrog Self-Hosted

   a. Enter the JFrog Artifactory URL as the Registry URL.

   For example, `https://artifactory.example.com/artifactory`, where `<artifactory.example.com>` is your server's domain or IP address.

   b. Under Authentication Method, enter your JFrog user credentials (Username and Password) for authentication.

   c. (Optional) Expand Show Advanced Settings.

   i. Select Use insecure connection to pull images if you want to allow image pull from the registry over an HTTP connection instead of HTTPS.

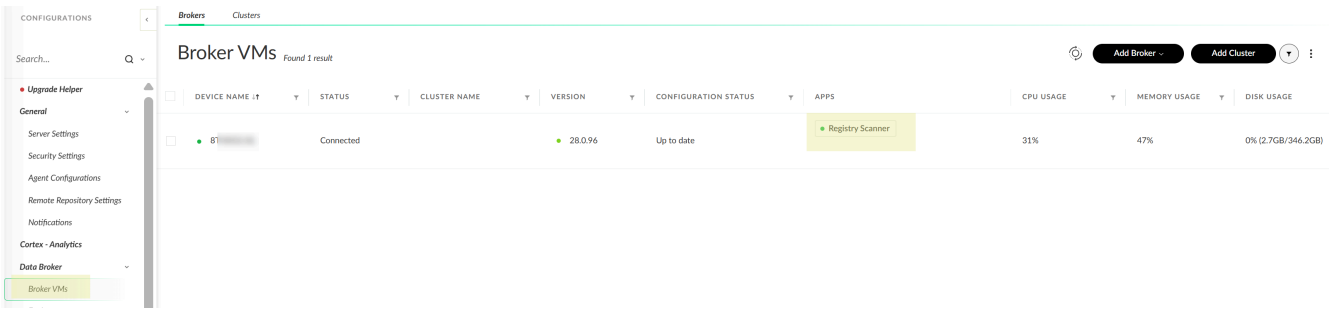
   ii. Enter the CA certificate if your JFrog Artifactory server uses a custom CA to sign the server certificate.
4. Select Next.

6. In Initial Scan Configuration, set your scanning process to focus on recently added or modified container images and exclude older ones that do not align with your current scanning objectives. This setting helps avoid unnecessary scans. Choose one of the following options:

   - All: Scans all container images, including all versions (tags), in all discovered repositories.
   - Latest Tag: Scans only images tagged 'latest' in all discovered repositories.
   - Days Modified: Scans container images created or modified in the last few days. You can select a range of up to 90 days for the scan.
7. Select Save.

   When the JFrog data source is saved successfully, a new data connector is created, and the initial discovery scan is started. The connection process may take up to 15 minutes. After the scan is complete, you can view the scanned details on the Container Images Inventory page. For more details, see Container Image assets.

   If you have selected the Scan with Broker VM option, then a Registry Scanner applet is created on the selected Broker VM or Cluster. For details, see Verify Registry Scanner connection.



After successfully adding a connector, go to the JFrog Artifactory Instances page to check the connector status and repository scan details.

You can also modify the connector settings and configure the scanning scope for images in the connected registry on this page.

To manage the connector, follow these steps:

1. Select Settings → Data Sources.
2. Find the JFrog instance from the list of 3rd Party Data Sources connectors, or use Search.
3. In the JFrog instance row, select View Details. The JFrog Artifactory Instances page appears.
4. On the JFrog Artifactory Instances page, you can filter results by any heading and value. You can also create a new instance by selecting + Add Instance and following the onboarding wizard to define the settings.
5. Select an instance name to open the details pane. The details pane contains the following granular information:

Instance Details	Description
Status	Shows the status of the connector: Connected, Error, Warning, Disabled, or Pending.
Applet Status on Broker VM	Shows the status of the Registry Scanner applet on the Broker VM page. This status is visible only when the Scan with Broker VM mode is selected.
Repositories	Shows the number of scanned repositories in the registry.
Scan Mode	Shows the selected scan mode for the data connector, such as Cloud Scan, Scan with Outpost, or Scan with Broker VM.
Security Capabilities	Shows a breakdown of the security capabilities enabled on the instance and their individual statuses. For example, select Registry Scanning when it shows a warning or error status to see the open errors and issues that contributed to the status.

6. You can also perform actions on each JFrog Artifactory instance: For example, select the  (pencil) icon to Edit the instance, or select the  (three dots) icon to Add image scope, Delete, or Disable the instance as follows:

Action	Instructions
Edit	Edit the JFrog Artifactory instance. **NOTE:** • If you selected Scan with Broker VM mode, you can't change to a different scan mode (such as Cloud Discovery or Scan with Outpost) when you edit the instance. • When editing an instance configured for Scan with Broker VM, you must re-enter your authentication credentials, including Username, Password, and CA certificate.
Add image scope	Define conditions to automatically include specific images in scanning. Conditions can be based on Repository or Tags. These conditions apply automatically to newly discovered images in the account.
Delete	Removes the connector.
Disable	Stops image scanning for the connector without deleting it.

1.6.6.8 | Cloud service provider permissions

Abstract

Grant the correct cloud service provider permissions for Cortex XSIAM.

When you set up Cortex XSIAM to collect data from your cloud environments, the onboarding wizard will ensure that the correct permissions are granted for Cortex XSIAM. The following tables list the permissions required for each of the options available in the onboarding wizards.

Review the permissions required for each cloud service provider:

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- Oracle Cloud Infrastructure

1.6.6.8.1 | Amazon Web Services provider permissions

Abstract

List of Amazon Web Services provider permissions for Cortex XSIAM.

ADS

Permission	Scope	Purpose
ec2:ModifySnapshotAttribute	• Snapshots with managed_by: paloaltonetworks tag • The snapshots can be shared only with the outpost account	Share snapshot with the outpost account
ec2:DeleteSnapshot	Snapshots with managed_by: paloaltonetworks tag	Delete scanned snapshot
ec2:CreateTags	Only as part of CreateSnapshot and CopySnapshot operations	Add tags for permission scoping and cost visibility
ec2:DescribeSnapshots	Snapshots with managed_by: paloaltonetworks tag	Retrieve snapshot creation status
ec2:CreateSnapshot	Snapshots created with managed_by: paloaltonetworks tag	Create disk snapshot
ec2:CopySnapshot	Snapshots copied with managed_by: paloaltonetworks tag	Re-encrypt snapshot with PANW's KMS key
kms:DescribeKey	• PANW's KMS key • Only EC2 Service can use this permission	To allow the re-encypt operation
kms:GenerateDataKeyWithoutPlaintext	• PANW's KMS key • Only EC2 Service can use this permission	To allow the re-encypt operation

Permission	Scope	Purpose
kms:CreateGrant	• PANW's KMS key • Only EC2 Service can use this permission	To allow the re-encypt operation

DSPM

Permission	Scope	Purpose
s3:List*	All S3 buckets	To allow the listing of all S3 objects
rds:DeleteDBSnapshot	PANW created snapshots	To delete snapshots created as part of the classification process
rds:AddTagsToResource	RDS resources in the account	Enables creating a unique tag for the created RDS resourceCreateDBSnapshots in order to find them at a later stage
rds:CancelExportTask	RDS resources in the account	Enables cancelling export tasks in case of failure or termination of the classification process
rds:CreateDBClusterSnapshot	RDS resources in the account	Enables creating a snapshot for the RDS clusters that need to be scanned at a later stage
rds:CreateDBSnapshot	RDS resources in the account	Enables creating a snapshot for the RDS instances that need to be scanned at a later stage
rds:Describe*	RDS resources in the account	Describe permissions enable PANW to get metadata information on the RDS instance
rds:List*	RDS resources in the account	List permissions enable PANW to understand which instances and snapshots exist in the account
rds:StartExportTask	RDS resources in the account	Enables to export data from the snapshots to an S3 bucket
s3:PutObject*	PANW created buckets	Enables writing data to an object in PANW's bucket to export data from the RDS instances
s3:DeleteObject*	PANW created buckets	Enables deleting stale objects that were created
s3:Get*	All S3 buckets	Enable PANW to read data within S3 buckets
kms:DescribeKey	KMS keys in the account	Enables getting information about the KMS keys in the account

Permission	Scope	Purpose
kms:GenerateDataKeyWithoutPlaintext	AWS account	Enables getting information about the KMS keys in the account
kms:CreateGrant	KMS keys in the account	The created EC2 instance sends a CreateGrant request to AWS KMS so that it can share the encrypted snapshot with the outpost account
iam:PassRole	PANW scanner role	Enables creating export tasks for RDS snapshots
arn:aws:iam::aws:policy/AmazonMemoryDBReadOnlyAccess	DynamoDB resource in the account	Read-only access to the MemoryDB resources
dynamodb:DescribeTable	All DynamoDB tables	Enables getting information about DynamoDB tables in the account
dynamodb:Scan	All DynamoDB tables	Enables accessing data in DynamoDB tables in the account
cloudwatch:GetMetricStatistics	All DynamoDB tables	Enables getting usage statistics, which is used to ensure that classification processes do not interfere with production environments

Discovery Engine

Permission	Purpose
arn:aws:iam::aws:policy/AmazonSQSReadOnlyAccess	Grants read-only access to Amazon Simple Queue Service (SQS). Allows the retrieval of SQS queue attributes, messages, and configurations.
arn:aws:iam::aws:policy/ReadOnlyAccess	Grants read-only access to AWS services and resources. Enables the ability to list and view configurations, metadata, and logs across AWS resources.
arn:aws:iam::aws:policy/SecurityAudit	Grants access to read security configuration metadata. Allows users to inspect IAM configurations, security policies, CloudTrail logs, and other security-relevant settings.
DS:DescribeDirectories	Grants read access to directory details in AWS Directory Service.
DS:ListTagsForResource	Lists tags associated with a specific AWS Directory Service resource
DirectConnect:DescribeConnections	Lists Direct Connect connections and their attributes
DirectConnect:DescribeDirectConnectGateways	Retrieves details about Direct Connect gateways
DirectConnect:DescribeVirtualInterfaces	Displays all virtual interfaces for an AWS account

Permission	Purpose
Glue:GetSecurityConfigurations	Retrieves security configurations for AWS Glue
WorkSpaces:DescribeTags	Lists tags associated with WorkSpaces resources
WorkSpaces:DescribeWorkspaceDirectories	Retrieves details about WorkSpaces directories
WorkSpaces:DescribeWorkspaces	Lists and describes WorkSpaces instances
apigateway:GetDomainNames	Retrieves API Gateway custom domain names
bedrock-agent:GetAgents	Retrieves details of Bedrock agents
bedrock-agent:GetDataSource	Retrieves details of a specific data source
bedrock-agent:GetKnowledgeBases	Retrieves details of knowledge bases
bedrock-agent:ListAgentAliases	Lists aliases associated with an agent
bedrock-agent:ListAgentKnowledgeBases	Lists knowledge bases linked to agents
bedrock-agent:ListAgents	Lists all Bedrock agents
bedrock-agent:ListDataSource	Lists available data sources
bedrock:ListCustomModel	Lists custom AI models in Amazon Bedrock, enabling visibility into custom AI model configurations
cloudcontrolapi:GetResource	Retrieves the state of an AWS resource managed via Cloud Control API
cloudformation:AmazonCloudFormation	General permission related to CloudFormation resource management
cloudformation:StackStatus	Retrieves the status of CloudFormation stacks
cloudformation:StackSummary	Provides a summary of CloudFormation stacks
cloudwatch:describeAlarms	Retrieves details about CloudWatch alarms
comprehendmedical:ListEntitiesDetectionV2Jobs	Lists entity detection jobs in Comprehend Medical
configservice:DescribeDeliveryChannels	Retrieves details of AWS Config delivery channels

Permission	Purpose
elasticfilesysttem:DescribeFileSystemPolicy	Retrieves policies associated with an EFS file system
elasticloadbalancingv2:DescribeSSLPolicies	Retrieves details of ELB SSL policies
forecast:ListTagsForResource	Lists tags associated with an Amazon Forecast resource
glue:GetConnections	Lists connection configurations for AWS Glue
glue:GetResourcePolicies	Retrieves Glue Data Catalog policies
iam:AmazonIdentityManagement	General IAM access for identity and access management
iam:AttachedPolicy	Retrieves policies attached to IAM identities
iam:PolicyRole	Lists IAM roles associated with a policy
iam:RoleDetail	Retrieves detailed information about IAM roles
opensearchserverless:ListCollections	Lists collections in OpenSearch Serverless
s3-control:GetAccessPointPolicy	Retrieves an S3 access point policy
s3-control:GetAccessPointPolicyStatus	Retrieves the status of an access point policy
s3-control:GetPublicAccessBlock	Retrieves the public access block configuration for an account
s3-control:ListAccessPoints	Lists S3 access points that are owned by the current account that's associated with the specified bucket
servicecatalog-appregistry:ListApplications	Lists applications in AWS AppRegistry
servicecatalog-appregistry:ListAttributeGroups	Lists attribute groups in AppRegistry

Registry Scan

Permission	Scope	Purpose
ecr:BatchGetImage	All ECR images in the account	Gets detailed information for an image, required in order to pull the image
ecr:GetDownloadUrlForLayer	All ECR images in the account	Used in the process of pulling images, to fetch the URL for the various layers that make up the image

Permission	Scope	Purpose
ecr:GetAuthorizationToken	All ECR images in the account	Used to create a login toker for pulling images from ECR
ecr-public:GetAuthorizationToken	All public ECR images in the account	Used to create a login token for pulling images from public ECR

Log Collection

Permission	Purpose
s3:GetObject	Grants permission to download objects from the configured S3 bucket
s3:ListBucket	Grants permission to see the specific bucket
sqs:ReceiveMessage	Grants permission to consume messages from the SQS queue to receive bucket notification messages
sqs:DeleteMessage	Grants permission to delete consumed messages, preventing re-processing of the same message
sqs:GetQueueAttributes	Grants permission to retrieve SQS queue attributes, used for metrics and monitoring

1.6.6.8.2 | Google Cloud Platform provider permissions

Abstract

List of Google Cloud Platform provider permissions for Cortex XSIAM.

ADS

Permission	Scope	Purpose
compute.snapshots.get	Snapshots with "cortex-scan-" prefix	Retrieve snapshot creation status
compute.snapshots.create	Snapshots with "cortex-scan-" prefix	Create disk snapshot
compute.snapshots.delete	Snapshots with "cortex-scan-" prefix	Delete scanned snapshot
compute.snapshots.setLabels	Snapshots with "cortex-scan-" prefix	Add snapshot labels for a cost visibility
compute.snapshots.useReadOnly	Snapshots with "cortex-scan-" prefix	Attach snapshot to a scanner VM

DSPM

Permission	Scope	Purpose	Notes
bigquery.bireservations.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.capacityCommitments.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.capacityCommitments.list	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.config.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.datasets.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.datasets.getIamPolicy	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.models.getData	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.models.getMetadata	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.models.list	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.routines.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.routines.list	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.tables.export	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	

Permission	Scope	Purpose	Notes
bigquery.tables.get	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.tables.getData	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.tables.getIamPolicy	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
bigquery.tables.list	All BigQuery instances	Enables classification of BigQuery by allowing access to data and usage	
cloudsql.backupRuns.get	All CloudSQL instances	Enables classification of CloudSQL by allowing access to data and backups	
cloudsql.backupRuns.create	All CloudSQL instances	Enables classification of CloudSQL by allowing access to data and backups	
cloudsql.backupRuns.delete	All CloudSQL instances	Enables classification of CloudSQL by allowing access to data and backups	
cloudsql.backupRuns.get	All CloudSQL instances	Enables classification of CloudSQL by allowing access to data and backups	
cloudsql.backupRuns.list	All CloudSQL instances	Enables classification of CloudSQL by allowing access to data and backups	
roles/cloudfunctions.viewer			Built-in role
roles/container.clusterViewer			Built-in role
roles/storage.objectViewer			Built-in role
roles/firebaserules.viewer			Built-in role

Discovery Engine

Permission	Purpose	Notes
roles/viewer		Built-in role
roles/cloudfunctions.viewer		Built-in role
roles/container.clusterViewer		Built-in role
roles/firebaserules.viewer		Built-in role
roles/storage.objectViewer		Built-in role
serviceusage.services.use	Use cloud services	
storage.buckets.get	Get metadata of a storage bucket	
storage.buckets.getIamPolicy	Get IAM policy of a storage bucket	
storage.buckets.list	List storage buckets	
storage.buckets.listEffectiveTags	List effective tags of storage buckets	
storage.buckets.listTagBindings	List tag bindings of storage buckets	
storage.objects.getIamPolicy	Get IAM policy of storage objects	
run.services.list	List Cloud Run services	
run.jobs.list	List Cloud Run jobs	
run.jobs.getIamPolicy	Get IAM policy of Cloud Run jobs	
cloudscheduler.jobs.list	List Cloud Scheduler jobs	
baremetalsolution.instances.list	List Bare Metal Solution instances	
baremetalsolution.networks.list	List Bare Metal Solution networks	
baremetalsolution.nfsshares.list	List Bare Metal Solution NFS shares	
baremetalsolution.volumes.list	List Bare Metal Solution volumes	

Permission	Purpose	Notes
baremetalsolution.luns.list	List Bare Metal Solution LUNs (Logical Unit Numbers)	
analyticshub.dataExchanges.list	List Analytics Hub data exchanges	
analyticshub.listings.getIamPolicy	Get IAM policy for Analytics Hub listings	
analyticshub.listings.list	List Analytics Hub listings	
notebooks.locations.list	List notebook locations	
notebooks.schedules.list	List notebook schedules	
composer.imageversions.list	List Composer image versions	
datamigration.connectionprofiles.list	List data migration connection profiles	
datamigration.connectionprofiles.getIamPolicy	Get IAM policy for data migration connection profiles	
datamigration.conversionworkspaces.list	List data migration conversion workspaces	
datamigration.conversionworkspaces.getIamPolicy	Get IAM policy for data migration conversion workspaces	
datamigration.migrationjobs.list	List data migration jobs	
datamigration.migrationjobs.getIamPolicy	Get IAM policy for data migration jobs	
datamigration.privateconnections.list	List data migration private connections	
datamigration.privateconnections.getIamPolicy	Get IAM policy for data migration private connections	
datamigration.migrationjobs.list	List AI Platform batch prediction jobs	
datamigration.migrationjobs.getIamPolicy	List AI Platform NAS jobs	
datamigration.privateconnections.list	List Cloud Security Scanner scans	
datamigration.privateconnections.getIamPolicy	Allows viewing the access policy for a Database Migration Service private connection	

Permission	Purpose	Notes
aiplatform.batchPredictionJobs.list	Allows listing AI Platform batch prediction jobs	
aiplatform.nasJobs.list	Allows listing AI Platform Neural Architecture Search (NAS) jobs	
cloudsecurityscanner.scans.list	Allows listing Cloud Security Scanner scans	

Log Collection

Permission	Purpose	Notes
roles/pubsub.subscriber	Grants access to consume messages from the subscription where audit logs are stored	Built-in role

Registry Scan

Permission	Scope	Purpose
artifactregistry.repositories.downloadArtifacts	All artifacts listed in the GAR of the customer's account	Needed in order to download images from GAR
roles/iam.serviceAccountTokenCreator	Access to this permission is limited to a specific Service Account defined within the Outpost. No account other than the defined Service Account can access the permission and access is limited to the permissions defined on the target SA	Allows impersonation to a specfiic service account

1.6.6.8.3 | Microsoft Azure provider permissions

Abstract

List of Microsoft Azure provider permissions for Cortex XSIAM.

ADS

Permission	Module	Scope	Purpose
Microsoft.Compute/snapshots/write	ADS	No scoping	Create disk snapshot
Microsoft.Compute/snapshots/delete	ADS	No scoping	Delete scanned snapshot
Microsoft.Compute/virtualMachines/read	ADS	No scoping	Allow disk snapshot operation
Microsoft.Compute/snapshots/read	ADS	No scoping	Convert snapshot to disk that will be attached to the scanner

DSPM

Permission	Scope	Purpose
Microsoft.Storage/storageAccounts/PrivateEndpointConnectionsApproval/action	Entire subscription	Enabling a scan by assigning private endpoints to a storage account located in a private network
Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read	All blobs	Enables classification of data in storage blobs
Microsoft.Storage/storageAccounts/fileServices/fileshares/files/read	All fileshares	Enables classification of data in storage fileshares
Microsoft.Storage/storageAccounts/listKeys/action	Entire subscription	Getting access key to the storage account to scan file share instances using API
Microsoft.Storage/storageAccounts/ListAccountSas/action	Entire subscription	Getting access SAS token to the storage account to scan file share instances using API
Microsoft.Storage/storageAccounts/PrivateEndpointConnectionsApproval/action	Entire subscription	Enabling a scan by assigning private endpoints to a storage account located in a private network
Microsoft.Storage/*/read	Entire subscription	Reading blobs data for data classification
Microsoft.Storage/storageAccounts/blobServices/generateUserDelegationKey/action	Entire subscription	Getting SAS token of blobServices to enable access
Microsoft.DocumentDB/databaseAccounts/listKeys/	Entire subscription	Getting SAS token of CosmosDB to enable access
Microsoft.Storage/storageAccounts/tableServices/tables/read	All storage tables	Enables classification of data in storage tables
Microsoft.CognitiveServices/*/read	All deployments	Enables discovery of OpenAI resources and other Azure AI services
Microsoft.CognitiveServices/*/action	All deployments	Enables reading and scanning OpenAI files and other Azure AI data resources
*/read	Entire subscription	Read-only access, used to get metadata of all managed data assets in the subscription
Microsoft.Network/routeTables/write	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/routeTables/join/action	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/routeTables/delete	PANW resources	Enables routing and managing internal traffic for DB classification

Permission	Scope	Purpose
Microsoft.Network/virtualNetworks/delete	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/virtualNetworks/join/action	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/virtualNetworks/subnets/delete	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/virtualNetworks/subnets/join/action	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/virtualNetworks/subnets/write	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/virtualNetworks/write	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/networkSecurityGroups/securityRules/write	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/networkSecurityGroups/securityRules/delete	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/networkSecurityGroups/join/action	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/networkSecurityGroups/delete	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Network/networkSecurityGroups/write	PANW resources	Enables routing and managing internal traffic for DB classification
Microsoft.Sql/servers/databases/read	PANW resources	Enables getting configurations on Azure SQL databases
Microsoft.Sql/servers/databases/write	PANW resources	Enables copying and managing SQL databases in Azure SQL server
Microsoft.Sql/servers/databases/resume/action	PANW resources	Enables copying and managing SQL databases in Azure SQL server
Microsoft.Sql/servers/databases/delete	PANW resources	Enable cleaning stale assets such as PANW's Azure SQL server databases
Microsoft.Sql/servers/delete	PANW resources	Enable cleaning stale assets such as PANW's Azure SQL server

Permission	Scope	Purpose
Microsoft.Sql/servers/write	PANW resources	Enables creating and managing PANW's Azure SQL servers
Microsoft.Sql/servers/virtualNetworkRules/write	PANW resources	Enables configuring network accessibility from the scanning VMs on PANW's Azure SQL servers
Microsoft.Sql/servers/privateEndpointConnectionsApproval/action	PANW resources	Enables connection using endpoints
Microsoft.Sql/managedInstances/write	PANW resources	Enables creation of SQL Managed Instance for classification of managed instances
Microsoft.Sql/managedInstances/databases/write	PANW resources	Used for copying PITR of SQL managed instances to PANW's resource group, enabling PANW to restore and scan it
Microsoft.Sql/managedInstances/delete	PANW resources	Enable cleaning stale assets such as PANW's Azure SQL Managed Instance

Discovery Engine

Permission	Purpose
Domain.Read.All	
EntitlementManagement.Read.All	
User.Read.All	
Policy.ReadWrite.AuthenticationMethod	
GroupMember.Read.All	
RoleManagement.Read.All	
Group.Read.All	
AuditLog.Read.All	
Policy.Read.All	
IdentityProvider.Read.All	
Directory.Read.All	

Permission	Purpose
Organization.Read.All	
Microsoft.ContainerInstance/containerGroups/containers/exec/action	Execute commands in a container
Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action	Get webhook callback config
Microsoft.DocumentDB/databaseAccounts/listConnectionStrings/action	List Cosmos DB connection strings
Microsoft.DocumentDB/databaseAccounts/listKeys/action	List Cosmos DB access keys
Microsoft.DocumentDB/databaseAccounts/readonlykeys/action	Get Cosmos DB read-only keys
Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action	View effective NSGs on NICs
Microsoft.Network/networkInterfaces/effectiveRouteTable/action	View effective route table on NICs
Microsoft.Network/networkWatchers/queryFlowLogStatus/*	Query NSG flow log status
Microsoft.Network/networkWatchers/read	Read network watcher settings
Microsoft.Network/networkWatchers/securityGroupView/action	View effective security rules
Microsoft.Network/virtualwans/vpnconfiguration/action	Download VPN config
Microsoft.Storage/storageAccounts/listKeys/action	List storage account keys
Microsoft.Web/sites/config/list/action	List web app configuration
Microsoft.Advisor/configurations/read	Read Advisor configuration
Microsoft.AlertsManagement/prometheusRuleGroups/read	Read Prometheus rule groups
Microsoft.AlertsManagement/smartDetectorAlertRules/read	Read smart detector alert rules
Microsoft.AnalysisServices/servers/read	Read Analysis Services servers

Permission	Purpose
Microsoft.ApiManagement/service/apis/diagnostics/read	Read diagnostics info of APIs
Microsoft.ApiManagement/service/apis/policies/read	Read policies on APIs
Microsoft.ApiManagement/service/apis/read	Read API details
Microsoft.ApiManagement/service/identityProviders/read	Read API Management identity providers
Microsoft.ApiManagement/service/portalsettings/read	Read developer portal settings
Microsoft.ApiManagement/service/products/policies/read	Read policies on API products
Microsoft.ApiManagement/service/products/read	Read API products
Microsoft.ApiManagement/service/read	Read API Management service info
Microsoft.ApiManagement/service/tenant/read	Read tenant info in API Management
Microsoft.AppConfiguration/configurationStores/read	Read Azure App Configuration stores
Microsoft.AppPlatform/Spring/apps/read	Read Spring apps in Azure App Platform
Microsoft.AppPlatform/Spring/read	Read Azure App Platform Spring resource info
Microsoft.Attestation/attestationProviders/read	Read attestation providers
Microsoft.Authorization/classicAdministrators/read	Read classic administrators info
Microsoft.Authorization/locks/read	Read resource locks
Microsoft.Authorization/permissions/read	Read permissions
Microsoft.Authorization/policyAssignments/read	Read policy assignments
Microsoft.Authorization/policyDefinitions/read	Read policy definitions

Permission	Purpose
Microsoft.Authorization/roleAssignments/read	Read role assignments
Microsoft.Authorization/roleDefinitions/read	Read role definitions
Microsoft.Automanage/configurationProfiles/Read	Read Automanage configuration profiles
Microsoft.Automation/automationAccounts/credentials/read	Read credentials in automation accounts
Microsoft.Automation/automationAccounts/hybridRunbookWorkerGroups/read	Read hybrid runbook worker groups
Microsoft.Automation/automationAccounts/read	Read automation accounts
Microsoft.Automation/automationAccounts/runbooks/read	Read runbooks
Microsoft.Automation/automationAccounts/variables/read	Read variables in automation accounts
Microsoft.AzureStackHCI/Clusters/Read	Read Azure Stack HCI clusters
Microsoft.Batch/batchAccounts/pools/read	Read batch account pools
Microsoft.Batch/batchAccounts/read	Read batch accounts
Microsoft.Blueprint/blueprints/read	Read blueprints
Microsoft.BotService/botServices/read	Read bot services
Microsoft.Cache/redis/firewallRules/read	Read firewall rules on Redis cache
Microsoft.Cache/redis/read	Read Redis caches
Microsoft.Cache/redisEnterprise/read	Read Redis Enterprise caches
Microsoft.Cdn/profiles/afdendpoints/read	Read CDN profile AFD endpoints
Microsoft.Cdn/profiles/afdendpoints/routes/read	Read routes of CDN profile AFD endpoints

Permission	Purpose
Microsoft.Cdn/profiles/customdomains/read	Read custom domains in CDN profiles
Microsoft.Cdn/profiles/endpoints/customdomains/read	Read custom domains of CDN endpoints
Microsoft.Cdn/profiles/endpoints/read	Read CDN profile endpoints
Microsoft.Cdn/profiles/origingroups/read	Read origin groups in CDN profiles
Microsoft.Cdn/profiles/read	Read CDN profiles
Microsoft.Cdn/profiles/securitypolicies/read	Read CDN profile security policies
Microsoft.Chaos/experiments/read	Read Chaos experiments
Microsoft.ClassicCompute/VirtualMachines/read	Read classic compute virtual machines
Microsoft.ClassicNetwork/networkSecurityGroups/read	Read classic network security groups
Microsoft.ClassicNetwork/reservedIps/read	Read classic network reserved IPs
Microsoft.ClassicNetwork/virtualNetworks/read	Read classic virtual networks
Microsoft.ClassicStorage/StorageAccounts/read	Read classic storage accounts
Microsoft.CognitiveServices/accounts/read	Read Cognitive Services accounts
Microsoft.CognitiveServices/accounts/deployments/read	Read deployments in Cognitive Services accounts
Microsoft.CognitiveServices/accounts/raiPolicies/read	Read RAI policies in Cognitive Services accounts
Microsoft.CognitiveServices/models/read	Read Cognitive Services models
Microsoft.CognitiveServices/accounts/models/read	Read models in Cognitive Services accounts

Permission	Purpose
Microsoft.Communication/CommunicationServices/Read	Read Communication Services
Microsoft.Compute/availabilitySets/read	Read availability sets
Microsoft.Compute/cloudServices/read	Read cloud services
Microsoft.Compute/cloudServices/roleInstances/read	Read cloud service role instances
Microsoft.Compute/diskEncryptionSets/read	Read disk encryption sets
Microsoft.Compute/disks/beginGetAccess/action	Begin get access on disks (action)
Microsoft.Compute/disks/read	Read disks
Microsoft.Compute/galleries/images/read	Read gallery images
Microsoft.Compute/galleries/read	Read galleries
Microsoft.Compute/hostGroups/read	Read host groups
Microsoft.Compute/snapshots/read	Read snapshots
Microsoft.Compute/virtualMachineScaleSets/networkInterfaces/read	Read network interfaces of VM scale sets
Microsoft.Compute/virtualMachineScaleSets/publicIPAddresses/read	Read public IP addresses of VM scale sets
Microsoft.Compute/virtualMachineScaleSets/read	Read virtual machine scale sets
Microsoft.Compute/virtualMachineScaleSets/virtualMachines/networkInterfaces/ipConfigurations/publicIPAddresses/read	Read public IPs of VM scale set VM NICs IP configs
Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read	Read virtual machines in VM scale sets
Microsoft.Compute/virtualMachineScaleSets/virtualmachines/instanceView/read	Read instance view of VM scale set VMs
Microsoft.Compute/virtualMachines/extensions/read	Read VM extensions

Permission	Purpose
Microsoft.Compute/virtualMachines/instanceView/read	Read VM instance view
Microsoft.Compute/virtualMachines/read	Read virtual machines
Microsoft.Confluent/organizations/Read	Read Confluent organizations
Microsoft.ContainerInstance/containerGroups/containers/exec/action	Execute commands in container instances
Microsoft.ContainerInstance/containerGroups/read	Read container groups
Microsoft.ContainerRegistry/registries/metadata/read	Read container registry metadata
Microsoft.ContainerRegistry/registries/pull/read	Read/pull from container registries
Microsoft.ContainerRegistry/registries/read	Read container registries
Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action	Get webhook callback config
Microsoft.ContainerService/managedClusters/read	Read managed Kubernetes clusters
Microsoft.DBforMariaDB/servers/firewallRules/read	Read MariaDB server firewall rules
Microsoft.DBforMariaDB/servers/read	Read MariaDB servers
Microsoft.DBforMySQL/flexibleServers/configurations/read	Read MySQL flexible server configurations
Microsoft.DBforMySQL/flexibleServers/databases/read	Read MySQL flexible server databases
Microsoft.DBforMySQL/flexibleServers/firewallRules/read	Read MySQL flexible server firewall rules
Microsoft.DBforMySQL/flexibleServers/read	Read MySQL flexible servers
Microsoft.DBforMySQL/servers/firewallRules/read	Read MySQL server firewall rules
Microsoft.DBforMySQL/servers/read	Read MySQL servers

Permission	Purpose
Microsoft.DBforMySQL/servers/virtualNetworkRules/read	Read MySQL server virtual network rules
Microsoft.DBforPostgreSQL/flexibleServers/configurations/read	Read PostgreSQL flexible server configurations
Microsoft.DBforPostgreSQL/flexibleServers/databases/read	Read PostgreSQL flexible server databases
Microsoft.DBforPostgreSQL/flexibleServers/firewallRules/read	Read PostgreSQL flexible server firewall rules
Microsoft.DBforPostgreSQL/flexibleServers/read	Read PostgreSQL flexible servers
Microsoft.DBforPostgreSQL/servers/configurations/read	Read PostgreSQL server configurations
Microsoft.DBforPostgreSQL/servers/firewallRules/read	Read PostgreSQL server firewall rules
Microsoft.DBforPostgreSQL/servers/read	Read PostgreSQL servers
Microsoft.DBforPostgreSQL/serversv2/firewallRules/read	Read PostgreSQL servers v2 firewall rules
Microsoft.Dashboard/grafana/read	Read Grafana dashboards
Microsoft.DataBoxEdge/dataBoxEdgeDevices/read	Read DataBox Edge devices
Microsoft.DataFactory/datafactories/read	Read Data Factory data factories
Microsoft.DataFactory/factories/integrationruntimes/read	Read Data Factory integration runtimes
Microsoft.DataFactory/factories/linkedservices/read	Read Data Factory linked services
Microsoft.DataFactory/factories/read	Read Data Factories
Microsoft.DataLakeAnalytics/accounts/dataLakeStoreAccounts/read	Read Data Lake Analytics associated Data Lake Store accounts

Permission	Purpose
Microsoft.DataLakeAnalytics/accounts/firewallRules/read	Read Data Lake Analytics firewall rules
Microsoft.DataLakeAnalytics/accounts/read	Read Data Lake Analytics accounts
Microsoft.DataLakeAnalytics/accounts/storageAccounts/read	Read Data Lake Analytics storage accounts
Microsoft.DataLakeStore/accounts/firewallRules/read	Read Data Lake Store firewall rules
Microsoft.DataLakeStore/accounts/read	Read Data Lake Store accounts
Microsoft.DataLakeStore/accounts/trustedIdProviders/read	Read Data Lake Store trusted ID providers
Microsoft.DataLakeStore/accounts/virtualNetworkRules/read	Read Data Lake Store virtual network rules
Microsoft.DataMigration/services/read	Read Data Migration services
Microsoft.DataShare/accounts/read	Read Data Share accounts
Microsoft.Databricks/accessConnectors/read	Read Databricks access connectors
Microsoft.Databricks/workspaces/read	Read Databricks workspaces
Microsoft.Datadog/monitors/read	Read Datadog monitors
Microsoft.DesktopVirtualization/applicationgroups/read	Read Desktop Virtualization application groups
Microsoft.DesktopVirtualization/hostpools/read	Read Desktop Virtualization host pools
Microsoft.DesktopVirtualization/hostpools/sessionhostconfigurations/read	Read Desktop Virtualization host pool session host configs
Microsoft.DesktopVirtualization/hostpools/sessionhosts/read	Read Desktop Virtualization host pool session hosts
Microsoft.DesktopVirtualization/workspaces/providers/Microsoft.Insights/diagnosticSettings/read	Read Desktop Virtualization workspace diagnostic settings

Permission	Purpose
Microsoft.DesktopVirtualization/workspaces/read	Read Desktop Virtualization workspaces
Microsoft.DevCenter/devcenters/read	Read DevCenter devcenters
Microsoft.DevTestLab/schedules/read	Read DevTestLab schedules
Microsoft.Devices/iotHubs/Read	Read IoT Hubs
Microsoft.Devices/iotHubs/privateLinkResources/Read	Read IoT Hubs private link resources
Microsoft.DigitalTwins/digitalTwinsInstances/read	Read Digital Twins instances
Microsoft.DocumentDB/cassandraClusters/read	Read DocumentDB Cassandra clusters
Microsoft.DocumentDB/databaseAccounts/listConnectionStrings/action	List connection strings of DocumentDB accounts (action)
Microsoft.DocumentDB/databaseAccounts/listKeys/action	List keys of DocumentDB accounts (action)
Microsoft.DocumentDB/databaseAccounts/read	Read DocumentDB database accounts
Microsoft.DocumentDB/databaseAccounts/readonlykeys/action	List readonly keys of DocumentDB accounts (action)
Microsoft.DomainRegistration/domains/Read	Read Domain registrations
Microsoft.Easm/workspaces/read	Read Easm workspaces
Microsoft.Elastic/monitors/read	Read Elastic monitors
Microsoft.EventGrid/domains/privateLinkResources/read	Read Event Grid domains private link resources
Microsoft.EventGrid/domains/read	Read Event Grid domains
Microsoft.EventGrid/namespaces/read	Read Event Grid namespaces

Permission	Purpose
Microsoft.EventGrid/partnerNamespaces/read	Read Event Grid partner namespaces
Microsoft.EventGrid/topics/privateLinkResources/read	Read Event Grid topics private link resources
Microsoft.EventGrid/topics/read	Read Event Grid topics
Microsoft.EventHub/Namespaces/PrivateEndpointConnections/read	Read EventHub Namespace private endpoint connections
Microsoft.EventHub/clusters/read	Read EventHub clusters
Microsoft.EventHub/namespaces/authorizationRules/read	Read EventHub namespaces authorization rules
Microsoft.EventHub/namespaces/eventhubs/authorizationRules/read	Read EventHub event hub authorization rules
Microsoft.EventHub/namespaces/eventhubs/read	Read EventHub event hubs
Microsoft.EventHub/namespaces/ipfilterrules/read	Read EventHub IP filter rules
Microsoft.EventHub/namespaces/read	Read EventHub namespaces
Microsoft.EventHub/namespaces/virtualnetworkrules/read	Read EventHub virtual network rules
Microsoft.HDInsight/clusters/applications/read	Read HDInsight cluster applications
Microsoft.HDInsight/clusters/read	Read HDInsight clusters
Microsoft.HealthBot/healthBots/Read	Read HealthBot bots
Microsoft.HealthcareApis/workspaces/read	Read Healthcare APIs workspaces
Microsoft.HybridCompute/machines/read	Read Hybrid Compute machines
Microsoft.Insights/ActivityLogAlerts/read	Read Insights activity log alerts

Permission	Purpose
Microsoft.Insights/Components/read	Read Insights components
Microsoft.Insights/DataCollectionEndpoints/Read	Read Insights data collection endpoints
Microsoft.Insights/DataCollectionRules/Read	Read Insights data collection rules
Microsoft.Insights/LogProfiles/read	Read Insights log profiles
Microsoft.Insights/MetricAlerts/Read	Read Insights metric alerts
Microsoft.Insights/actionGroups/read	Read Insights action groups
Microsoft.Insights/diagnosticSettings/read	Read Insights diagnostic settings
Microsoft.Insights/eventtypes/values/read	Read Insights event type values
Microsoft.IoTCentral/IoTApps/read	Read IoT Central applications
Microsoft.KeyVault/vaults/keys/read	Read Key Vault keys
Microsoft.KeyVault/vaults/privateLinkResources/read	Read Key Vault private link resources
Microsoft.KeyVault/vaults/read	Read Key Vault vaults
Microsoft.Kusto/Clusters/Databases/read	Read Kusto cluster databases
Microsoft.Kusto/Clusters/read	Read Kusto clusters
Microsoft.Kusto/clusters/read	Read Kusto clusters (alternative)
Microsoft.LabServices/labs/read	Read Lab Services labs
Microsoft.LoadTestService/loadTests/read	Read Load Test Service tests
Microsoft.Logic/integrationAccounts/read	Read Logic integration accounts

Permission	Purpose
Microsoft.Logic/workflows/read	Read Logic workflows
Microsoft.Logic/workflows/versions/read	Read Logic workflow versions
Microsoft.MachineLearningServices/workspaces/computes/read	Read Machine Learning Services workspace computes
Microsoft.MachineLearningServices/workspaces/outboundRules/read	Read Machine Learning Services workspace outbound rules
Microsoft.MachineLearningServices/workspaces/read	Read Machine Learning Services workspaces
Microsoft.ManagedIdentity/userAssignedIdentities/read	Read Managed Identity user assigned identities
Microsoft.ManagedServices/marketplaceRegistrationDefinitions/read	Read Managed Services marketplace registration defs
Microsoft.ManagedServices/registrationAssignments/read	Read Managed Services registration assignments
Microsoft.Management/managementGroups/descendants/read	Read Management Groups descendants
Microsoft.Management/managementGroups/read	Read Management Groups
Microsoft.Management/managementGroups/subscriptions/read	Read Management Groups subscriptions
Microsoft.Maps/accounts/read	Read Maps accounts
Microsoft.Migrate/moveCollections/read	Read Migrate move collections
Microsoft.MixedReality/ObjectAnchorsAccounts/read	Read Mixed Reality Object Anchors accounts
Microsoft.NetApp/netAppAccounts/capacityPools/read	Read NetApp capacity pools
Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read	Read NetApp volumes
Microsoft.NetApp/netAppAccounts/read	Read NetApp accounts

Permission	Purpose
Microsoft.Network/ApplicationGatewayWebApplicationFirewallPolicies/read	Read Application Gateway Web Application Firewall Policies
Microsoft.Network/applicationGateways/read	Read Application Gateways
Microsoft.Network/applicationSecurityGroups/read	Read Application Security Groups
Microsoft.Network/azurefirewalls/read	Read Azure Firewalls
Microsoft.Network/bastionHosts/read	Read Bastion Hosts
Microsoft.Network/connections/read	Read Network Connections
Microsoft.Network/ddosProtectionPlans/read	Read DDoS Protection Plans
Microsoft.Network/dnsZones/read	Read DNS Zones
Microsoft.Network/expressRouteCircuits/authorizations/read	Read ExpressRoute Circuit authorizations
Microsoft.Network/expressRouteCircuits/peerings/connections/read	Read ExpressRoute Circuit peerings connections
Microsoft.Network/expressRouteCircuits/peerings/peerConnections/read	Read ExpressRoute Circuit peer connections
Microsoft.Network/expressRouteCircuits/peerings/read	Read ExpressRoute Circuit peerings
Microsoft.Network/expressRouteCircuits/read	Read ExpressRoute Circuits
Microsoft.Network/expressRouteCrossConnections/peerings/read	Read ExpressRoute Cross Connections peerings
Microsoft.Network/expressRouteCrossConnections/read	Read ExpressRoute Cross Connections
Microsoft.Network/expressRouteGateways/expressRouteConnections/read	Read ExpressRoute Gateways connections
Microsoft.Network/expressRouteGateways/read	Read ExpressRoute Gateways

Permission	Purpose
Microsoft.Network/expressRoutePorts/authorizations/read	Read ExpressRoute Ports authorizations
Microsoft.Network/expressRoutePorts/links/read	Read ExpressRoute Ports links
Microsoft.Network/expressRoutePorts/read	Read ExpressRoute Ports
Microsoft.Network/expressRoutePortsLocations/read	Read ExpressRoute Ports locations
Microsoft.Network/firewallPolicies/read	Read Firewall Policies
Microsoft.Network/frontDoorWebApplicationFirewallPolicies/read	Read Front Door Web Application Firewall Policies
Microsoft.Network/frontDoors/backendPools/read	Read Front Door backend pools
Microsoft.Network/frontDoors/frontendEndpoints/read	Read Front Door frontend endpoints
Microsoft.Network/frontDoors/healthProbeSettings/read	Read Front Door health probe settings
Microsoft.Network/frontDoors/loadBalancingSettings/read	Read Front Door load balancing settings
Microsoft.Network/frontDoors/read	Read Front Doors
Microsoft.Network/frontDoors/routingRules/read	Read Front Door routing rules
Microsoft.Network/frontDoors/rulesEngines/read	Read Front Door rules engines
Microsoft.Network/loadBalancers/read	Read Load Balancers
Microsoft.Network/localnetworkgateways/read	Read Local Network Gateways
Microsoft.Network/locations/usages/read	Read Network locations usage
Microsoft.Network/natGateways/read	Read NAT Gateways

Permission	Purpose
Microsoft.Network/networkInterfaces/effectiveNetworkSecurityGroups/action	Execute effective network security groups action
Microsoft.Network/networkInterfaces/effectiveRouteTable/action	Execute effective route table action
Microsoft.Network/networkInterfaces/read	Read Network Interfaces
Microsoft.Network/networkSecurityGroups/defaultSecurityRules/read	Read Network Security Groups default security rules
Microsoft.Network/networkSecurityGroups/read	Read Network Security Groups
Microsoft.Network/networkSecurityGroups/securityRules/read	Read Network Security Groups security rules
Microsoft.Network/networkWatchers/queryFlowLogStatus/*	Query network watcher flow log status
Microsoft.Network/networkWatchers/read	Read Network Watchers
Microsoft.Network/networkWatchers/securityGroupView/action	Execute security group view action
Microsoft.Network/p2sVpnGateways/read	Read P2S VPN Gateways
Microsoft.Network/privateDnsZones/ALL/read	Read Private DNS Zones ALL
Microsoft.Network/privateDnsZones/read	Read Private DNS Zones
Microsoft.Network/privateEndpoints/privateDnsZoneGroups/read	Read Private Endpoints DNS Zone Groups
Microsoft.Network/privateEndpoints/read	Read Private Endpoints
Microsoft.Network/privateLinkServices/read	Read Private Link Services
Microsoft.Network/publicIPAddresses/read	Read Public IP Addresses
Microsoft.Network/publicIPPrefixes/read	Read Public IP Prefixes
Microsoft.Network/routeFilters/read	Read Route Filters

Permission	Purpose
Microsoft.Network/routeFilters/routeFilterRules/read	Read Route Filter Rules
Microsoft.Network/routeTables/read	Read Route Tables
Microsoft.Network/routeTables/routes/read	Read Route Table Routes
Microsoft.Network/serviceEndpointPolicies/read	Read Service Endpoint Policies
Microsoft.Network/serviceEndpointPolicies/serviceEndpointPolicyDefinitions/read	Read Service Endpoint Policy Definitions
Microsoft.Network/trafficManagerProfiles/read	Read Traffic Manager Profiles
Microsoft.Network/virtualNetworkGateways/read	Read Virtual Network Gateways
Microsoft.Network/virtualNetworks/read	Read Virtual Networks
Microsoft.Network/virtualNetworks/subnets/read	Read Virtual Network Subnets
Microsoft.Network/virtualNetworks/virtualNetworkPeerings/read	Read Virtual Network Peerings
Microsoft.Network/virtualWans/read	Read Virtual WANs
Microsoft.Network/virtualwans/vpnconfiguration/action	Execute VPN configuration action
Microsoft.Network/vpnServerConfigurations/read	Read VPN Server Configurations
Microsoft.NetworkFunction/azureTrafficCollectors/read	Read Azure Traffic Collectors
Microsoft.NotificationHubs/Namespaces/NotificationHubs/read	Read Notification Hubs
Microsoft.NotificationHubs/Namespaces/read	Read Notification Hub namespaces
Microsoft.OperationalInsights/clusters/read	Read Operational Insights clusters
Microsoft.OperationalInsights/querypacks/read	Read Operational Insights query packs

Permission	Purpose
Microsoft.OperationalInsights/workspaces/read	Read Operational Insights workspaces
Microsoft.OperationalInsights/workspaces/tables/read	Read Operational Insights workspace tables
Microsoft.Orbital/spacecrafts/read	Read Orbital spacecrafts
Microsoft.PowerBIDedicated/capacities/read	Read Power BI Dedicated capacities
Microsoft.PowerBIDedicated/servers/read	Read Power BI Dedicated servers
Microsoft.Quantum/Workspaces/Read	Read Quantum Workspaces
Microsoft.RecoveryServices/Vaults/backupProtectedItems/read	Read Recovery Services Vault backup protected items
Microsoft.RecoveryServices/Vaults/read	Read Recovery Services Vaults
Microsoft.RecoveryServices/vaults/backupPolicies/read	Read Recovery Services Vault backup policies
Microsoft.RedHatOpenShift/openShiftClusters/read	Read Red Hat OpenShift clusters
Microsoft.Relay/Namespaces/read	Read Relay namespaces
Microsoft.Resources/Resources/read	Read generic resources
Microsoft.Resources/subscriptions/providers/read	Read subscription providers
Microsoft.Resources/subscriptions/read	Read subscriptions
Microsoft.Resources/subscriptions/resourceGroups/read	Read resource groups
Microsoft.Resources/subscriptions/resourceGroups/write	Write resource groups
Microsoft.Resources/templateSpecs/read	Read template specs
Microsoft.SaaS/applications/read	Read SaaS applications

Permission	Purpose
Microsoft.Search/searchServices/read	Read Azure Search services
Microsoft.Security/advancedThreatProtectionSettings/read	Read Security advanced threat protection settings
Microsoft.Security/autoProvisioningSettings/read	Read Security auto provisioning settings
Microsoft.Security/automations/read	Read Security automations
Microsoft.Security/iotSecuritySolutions/read	Read IoT Security Solutions
Microsoft.Security/locations/jitNetworkAccessPolicies/read	Read Just-in-Time network access policies
Microsoft.Security/locations/read	Read Security locations
Microsoft.Security/pricings/read	Read Security pricings
Microsoft.Security/secureScores/read	Read Security secure scores
Microsoft.Security/securityContacts/read	Read Security contacts
Microsoft.Security/settings/read	Read Security settings
Microsoft.Security/workspaceSettings/read	Read Security workspace settings
Microsoft.ServiceBus/namespaces/authorizationRules/read	Read Service Bus namespace authorization rules
Microsoft.ServiceBus/namespaces/networkrulesets/read	Read Service Bus namespace network rule sets
Microsoft.ServiceBus/namespaces/privateEndpointConnections/read	Read Service Bus namespace private endpoint connections
Microsoft.ServiceBus/namespaces/providers/Microsoft.Insights/diagnosticSettings/read	Read Service Bus namespace diagnostic settings
Microsoft.ServiceBus/namespaces/queues/read	Read Service Bus queues
Microsoft.ServiceBus/namespaces/read	Read Service Bus namespaces

Permission	Purpose
Microsoft.ServiceBus/namespaces/topics/read	Read Service Bus topics
Microsoft.ServiceBus/namespaces/topics/subscriptions/read	Read Service Bus topic subscriptions
Microsoft.ServiceFabric/clusters/read	Read Service Fabric clusters
Microsoft.SignalRService/SignalR/read	Read SignalR Service SignalR
Microsoft.SignalRService/WebPubSub/read	Read SignalR Web PubSub
Microsoft.Solutions/applications/read	Read Solutions applications
Microsoft.Sql/managedInstances/databases/read	Read SQL managed instances databases
Microsoft.Sql/managedInstances/databases/transparentDataEncryption/read	Read SQL managed instances databases Transparent Data Encryption
Microsoft.Sql/managedInstances/encryptionProtector/Read	Read SQL managed instances encryption protector
Microsoft.Sql/managedInstances/read	Read SQL managed instances
Microsoft.Sql/managedInstances/vulnerabilityAssessments/Read	Read SQL managed instances vulnerability assessments
Microsoft.Sql/servers/administrators/read	Read SQL server administrators
Microsoft.Sql/servers/auditingSettings/read	Read SQL server auditing settings
Microsoft.Sql/servers/databases/auditingSettings/read	Read SQL server databases auditing settings
Microsoft.Sql/servers/databases/dataMaskingPolicies/read	Read SQL server databases data masking policies
Microsoft.Sql/servers/databases/dataMaskingPolicies/rules/read	Read SQL server databases data masking policies rules
Microsoft.Sql/servers/databases/read	Read SQL server databases

Permission	Purpose
Microsoft.Sql/servers/databases/securityAlertPolicies/read	Read SQL server databases security alert policies
Microsoft.Sql/servers/databases/transparentDataEncryption/read	Read SQL server databases Transparent Data Encryption
Microsoft.Sql/servers/encryptionProtector/read	Read SQL server encryption protector
Microsoft.Sql/servers/firewallRules/read	Read SQL server firewall rules
Microsoft.Sql/servers/read	Read SQL servers
Microsoft.Sql/servers/securityAlertPolicies/read	Read SQL server security alert policies
Microsoft.Sql/servers/vulnerabilityAssessments/read	Read SQL server vulnerability assessments
Microsoft.SqlVirtualMachine/sqlVirtualMachines/read	Read SQL Virtual Machines
Microsoft.Storage/storageAccounts/blobServices/read	Read Storage blob services
Microsoft.Storage/storageAccounts/fileServices/read	Read Storage file services
Microsoft.Storage/storageAccounts/fileServices/shares/read	Read Storage file shares
Microsoft.Storage/storageAccounts/listKeys/action	List Storage account keys (action)
Microsoft.Storage/storageAccounts/providers/Microsoft.Insights/diagnosticSettings/read	Read Storage account diagnostic settings
Microsoft.Storage/storageAccounts/queueServices/read	Read Storage queue services
Microsoft.Storage/storageAccounts/read	Read Storage accounts
Microsoft.Storage/storageAccounts/tableServices/read	Read Storage table services
Microsoft.StorageCache/Subscription/caches/read	Read Storage Cache subscription caches
Microsoft.StorageCache/caches/read	Read Storage Cache caches

Permission	Purpose
Microsoft.StorageMover/storageMovers/read	Read Storage Mover storage movers
Microsoft.StorageSync/storageSyncServices/privateLinkResources/read	Read Storage Sync private link resources
Microsoft.StorageSync/storageSyncServices/read	Read Storage Sync services
Microsoft.StreamAnalytics/clusters/Read	Read Stream Analytics clusters
Microsoft.StreamAnalytics/streamingjobs/Read	Read Stream Analytics streaming jobs
Microsoft.Subscription/Policies/default/read	Read Subscription default policies
Microsoft.Synapse/privateLinkHubs/privateLinkResources/read	Read Synapse private link hubs private link resources
Microsoft.Synapse/privateLinkHubs/read	Read Synapse private link hubs
Microsoft.Synapse/workspaces/privateLinkResources/read	Read Synapse workspace private link resources
Microsoft.Synapse/workspaces/read	Read Synapse workspaces
Microsoft.Synapse/workspaces/sparkConfigurations/read	Read Synapse workspaces spark configurations
Microsoft.Synapse/workspaces/sqlPools/geoBackupPolicies/read	Read Synapse workspaces SQL pools geo backup policies
Microsoft.Synapse/workspaces/sqlPools/read	Read Synapse workspaces SQL pools
Microsoft.VideoIndexer/accounts/read	Read Video Indexer accounts
Microsoft.VisualStudio/Account/Read	Read Visual Studio accounts
Microsoft.Web/certificates/read	Read Web certificates
Microsoft.Web/customApis/read	Read Web custom APIs

Permission	Purpose
Microsoft.Web/hostingEnvironments/Read	Read Web hosting environments
Microsoft.Web/serverfarms/Read	Read Web server farms
Microsoft.Web/sites/Read	Read Web sites
Microsoft.Web/sites/basicPublishingCredentialsPolicies/Read	Read Web sites basic publishing credentials policies
Microsoft.Web/sites/config/list/action	Execute action to list Web sites config
Microsoft.Web/sites/config/read	Read Web sites config
Microsoft.web/sites/config/appsettings/read	Read Web sites app settings
Microsoft.Web/sites/privateEndpointConnections/Read	Read Web sites private endpoint connections
Microsoft.Web/sites/read	Read Web sites
Microsoft.Web/sites/slots/Read	Read Web sites slots
microsoft.web/serverfarms/sites/read	Read Server farms sites
Microsoft.Web/staticSites/Read	Read Web static sites
Microsoft.Workloads/monitors/read	Read Workloads monitors
Microsoft.classicCompute/domainNames/read	Read Classic Compute domain names
microsoft.app/containerapps/read	Read App container apps
microsoft.monitor/accounts/read	Read Monitor accounts
microsoft.network/virtualnetworkgateways/connections/read	Read Virtual network gateways connections

Log Collection

Permission	Scope	Purpose
Azure Event Hubs Data Receiver	Event Hub that was created during the onboarding (containing the audit logs)	Allows receive access to Azure Event Hubs resources
Storage Blob Data Contributor	Storage account that was created during the onboarding	Reads, writes, and deletes Azure Storage containers and blobs

Registry Scan

Permission
Microsoft.ContainerRegistry/registries/metadata/read
Microsoft.ContainerRegistry/registries/pull/read
Microsoft.ContainerRegistry/registries/read
Microsoft.ContainerRegistry/registries/webhooks/getCallbackConfig/action

1.6.6.8.4 | Oracle Cloud Infrastructure provider permissions

Abstract

List of Oracle Cloud Infrastructure provider permissions for Cortex XSIAM.

ADS

Permission	Module	Scope	Purpose
Admit group CortexOutpostGroup of tenancy CortexOutpost to use volumes in tenancy	ADS	In tenancy	Allow creation of backups from volumes
Admit group CortexOutpostGroup of tenancy CortexOutpost to use key-delegate in tenancy	ADS	In tenancy	Re-encrypt backups during copy/restore operations
Admit group CortexOutpostGroup of tenancy CortexOutpost to associate keys in tenancy with volumes in tenancy CortexOutpost	ADS	Volumes in tenancy	Associate encryption keys with volumes during backup/restore
Admit group CortexOutpostGroup of tenancy CortexOutpost to use tag-namespaces in tenancy	ADS	In tenancy	Enable tagging for permission scoping, resource tracking, and cost visibility
Admit group CortexOutpostGroup of tenancy CortexOutpost to manage boot-volume-backups in tenancy where request.operation != 'DeleteBootVolumeBackup'	ADS	Excludes delete	Allow full management of boot volume backups except deletion

Permission	Module	Scope	Purpose
Admit group CortexOutpostGroup of tenancy CortexOutpost to manage boot-volume-backups in tenancy where target.resource.tag.cortex_m-o-lcaas_id.panw_capability = 'cortex-scan-platform'	ADS	Only boot-volume-backups tagged with panw_capability = cortex-scan-platform	Restrict deletion to Cortex scan-related resources only
Admit group CortexOutpostGroup of tenancy CortexOutpost to read all-resources in tenancy	ADS	In tenancy	Read-only access to all resources

Discovery Engine

"Discovery Engine" read only access. Grants read-only access to OCI tenancy and resources.

1.6.7 | Ingest data for API security

Abstract

Ingest API data to analyze and identify potential security threats.

Configure the settings in both Cortex XSIAM and your cloud service provider to retrieve and collect API data for further analysis by Cortex's comprehensive API security capabilities that provides a transparent view of API traffic, helping to identify potential security threats.

1.6.7.1 | Ingest AWS API Gateway

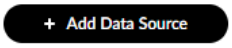
Abstract

Integrate Cortex with AWS API gateway to scan APIs for threats and vulnerabilities.

Integrate AWS API Gateway with Cortex XSIAM to begin scanning the APIs for potential threats and vulnerabilities.

Settings in Cortex XSIAM

In Cortex XSIAM, set up the AWS API Gateway data source to integrate with the AWS API Gateway.

1. From Settings → Data Sources , click 

and search for AWS API Gateway and then click Connect or Connect Another Instance.
2. In the AWS API Collector wizard, enter a relevant name and click Create and Proceed.
3. Copy the key and save it for later.

   > **NOTE:**
   >
   > You must generate a new key if you did not save.

4. Click Close.
5. An instance is created for the data source. Click  (Copy API URL) to establish a connection when setting up AWS API Gateway.

Settings in AWS Management Console

Configure the settings in the AWS Management Console to integrate with Cortex XSIAM:

1. Log in to the AWS Management Console.
2. In AWS Management Console, navigate to API Gateway.
   a. Expand the left-hand menu of the API project.
   b. Go to Settings → Logging and click Edit. Verify that the CloudWatch log role ARN is filled.
   c. Click Stages and from Stages, select the relevant stage.
   d. From Logs and Tracing, click Edit and configure the following:

- CloudWatch Logs: Select Errors and info logs
- Select Data tracing
- Select Detailed metrics

e. Click Save.

This creates a unique log group inside CloudWatch.

3. Open CloudWatch in another window by typing CloudWatch in the search bar.

a. Go to Logs → Log groups and search for the log group just created.

The group name follows the following format: “`API-Gateway-Execution-Logs_<gw ID>/<stage name>`”

b. Click the log group, and from the Log group details, copy the ARN.

4. Return to Edit logs and tracing, go to enable the custom access logging , and paste the ARN without the * in the Access log destination ARN field.

Example 161.

ARN: `arn:aws:logs:us-east-1:123456789012:log-group:API-Gateway-Execution-Logs_153tp249k2/Prod:*`

Paste in Access log destination ARN: `arn:aws:logs:us-east-1:123456789012:log-group:API-Gateway-Execution-Logs_153tp249k2/Prod`

5. In Log format, type the following and click Save:

```
($context.requestId) accountId: $context.accountId; requestTime: $context.requestTime; path: $context.path
```

6. Open Firehose in another window by typing Firehose in the search bar.

a. Configure the following:

- Source: Direct PUT
- Destination: HTTP Endpoint
- Firehose stream name: Add a relevant name.

b. In Destination settings, configure the following:

- HTTP endpoint URL : Add the API URL from Cortex XSIAM.
- Authentication: Select Use access key.
- Access key: Paste the generated token from AWS API Gateway.
- Content encoding: Select GZIP.

c. In Backup settings, configure the following:

- Source record backup in Amazon S3: select Failed date only.
- S3 backup bucket: select a bucket or enter a bucket URI.

d. Click Create.

It takes up to 5 minutes for the stream to be activated.

7. Refer to Subscription filters with Amazon Data Firehose. To create an IAM Role and provide CloudWatch with the appropriate permissions for the streaming, refer to steps 8-12.

```
aws logs put-subscription-filter \
    --log-group-name "<YOUR_LOG_GROUP_NAME>" \
    --filter-name "<any_filter_name>" \
    --filter-pattern "" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/<YOUR_DELIVERY_STREAM>" \
    --role-arn "arn:aws:iam::<ACCOUNT_ID>:role/<YOUR_IAM_ROLE>"
```

> **IMPORTANT:**
>
> Leave `–filter-pattern` empty as displayed above.

After the Data Firehose delivery stream is active and you have created the IAM role, you can create the CloudWatch Logs subscription filter. The subscription filter immediately starts the flow of real-time log data from the chosen log group to your Amazon Data Firehose delivery stream.

After you create the filter, go back to Data Sources → AWS API Gateway to see the logs starting to come in.

> **NOTE:**

> If no logs are showing, send some API requests on Postman or CURL.

1.6.7.2 | Ingest Azure APIM

Abstract

Integrate Cortex with Azure APIM to scan APIs for threats and vulnerabilities.

> **LICENSE TYPE:**
>
> Requires the Data Collection add-on.

Integrate Azure APIM with Cortex XSIAM to start scanning its APIs for potential threats and vulnerabilities.

You need to set up a policy that enables you to customize the behavior of managed APIs. You can configure the sending of HTTP request/response data to Cortex XSIAM. The data is saved and analyzed by API security modules, which provide information on the security risks associated with the APIs.

> **NOTE:**
>
> Microsoft Azure APIM service must be running before starting to configure the integration.

Settings in Cortex XSIAM

In Cortex XSIAM, set up the Azure API Management data source to integrate with the Azure API Gateway.

1. From Settings → Data Sources , click 

and search for Azure API Management and click Connect or Connect Another Instance.
2. In the APIM Collector wizard, enter a relevant name and then click Create and Proceed.
3. Copy the key and paste it somewhere so that you can access it for later.

   If you forget to record the key and close the window, you must generate a new key and repeat this process.
4. Click Close.
5. An instance is created for the data source. Click  (Copy API URL) to establish a connection when setting up Azure APIM.

Settings in Azure APIM policy

Configure an inbound and outbound policy to send HTTP traffic data of the APIs to Cortex XSIAM. You can configure a policy for individual operations (endpoints) or all operations of a single API.

Follow the steps to configure the policy.

1. Log in to Microsoft Azure.
2. Go to API Management services and select the relevant service.
3. From the left-hand menu, select APIs → Named values.

   > **NOTE:**
   >
   > From the URL, save the UUID and the resource group - `/resource/subscriptions/<UUID>/resourceGroups/<ResourceGroup>`.
   >
   > The UUID is the Azure account/subscription ID and the resource group, which is the group where the APIM Service is defined.
4. Configure the settings in each of the sections. Follow the steps in the order they are listed.

   > **NOTE:**
   >
   > Use the search to navigate to the specific section.

   Named values: Add the values:

- cloud-account-id
    - Type: Plain
    - Value: The UUID you saved from the previous step.
- cloud-resource-group
    - Type: Plain
    - Value: The resource group you saved from the previous step.
- cortex-api-key
    - Type: Secret
    - Value: The token that you saved from data sources in Cortex.
- cortex-api-url
    - Type: Plain
    - Value: The API URL from data sources in Cortex.
- cortex-http-body-size-limit-bytes
    - Type: Plain
    - Value: 131072

    > **NOTE:**
    >
    > 131072 bytes = 128 KB. This value determines the size (in bytes) of request and response bodies to send to Cortex. Any bytes beyond this limit are truncated.

APIs: From the left-hand menu, go to APIs → APIs.

1. You can create a policy on a specific API or choose to create a policy on all APIs.
2. From Inbound Processing, click </> .

   The Policies screen opens. There are three sections:

   - `<inbound>`
   - `<backend>`
   - `<outbound>`

   The `<inbound>` includes the request before it's sent to the `<outbound>`. The parameters are saved before they're sent.

   Add the following inside the `<inbound>`:

```
 <!-- Save the request body and headers to be sent to Cortex. This should always be placed at the very beginning of the inbound element. -->
        <set-variable name="requestBody" value="@((context.Request?.Body?.As<string>(preserveContent: true)) ?? string.Empty)" />
        <set-variable name="requestHeaders" value="@(JsonConvert.SerializeObject(context.Request.Headers))" />
        <!-- End of setting variables for sending to Cortex --><!-- Save the request body and headers to be sent to Cortex. This should always be placed at the
very beginning of the inbound element. -->
        <set-variable name="requestBody" value="@((context.Request?.Body?.As<string>(preserveContent: true)) ?? string.Empty)" />
        <set-variable name="requestHeaders" value="@(JsonConvert.SerializeObject(context.Request.Headers))" />
        <!-- End of setting variables for sending to Cortex -->
```

   > **NOTE:**
   >
   > If any other inbound policies should be added, they must be added after these elements.

   The `<outbound>` includes the request before it returns a response.

   Add the following inside the <outbound> element, at the end, after the other child elements:

```
<!-- Send data to Cortex. This should always be placed at the very end of the outbound element. -->
        <send-request mode="new" response-variable-name="mirrorMessage">
            <set-url>{{cortex-api-url}}</set-url>
            <set-method>POST</set-method>
            <set-header name="Content-Type" exists-action="override">
                <value>application/json</value>
            </set-header>
            <set-header name="Authorization" exists-action="override">
                <value>{{cortex-api-key}}</value>
            </set-header>
            <set-body>@{
                        string requestBody = context.Variables.GetValueOrDefault<string>("requestBody");
                        string responseBody = context.Response.Body.As<string>(preserveContent: true);
                        int bodySizeLimit = {{cortex-http-body-size-limit-bytes}};
                        bool requestBodySizeExceedsLimit = requestBody.Length > bodySizeLimit;
```

```
                bool responseBodySizeExceedsLimit = responseBody.Length > bodySizeLimit;

                return JsonConvert.SerializeObject(new {
                    // Resource information
                    subscriptionID          = "{{cloud-account-id}}",
                    resourceGroup           = "{{cloud-resource-group}}",
                    serviceID               = context.Deployment.ServiceId,
                    region                  = context.Deployment.Region,
                    apiID                   = context.Api.Id,
                    apiRevision             = context.Api.Revision,
                    // Request information
                    requestID               = context.RequestId,
                    url                     = context.Request.OriginalUrl,
                    httpMethod              = context.Request.Method,
                    requestBody             = requestBodySizeExceedsLimit ? requestBody.Substring(0, bodySizeLimit) : requestBody,
                    requestBodyTruncated    = requestBodySizeExceedsLimit,
                    requestHeaders          = JsonConvert.DeserializeObject(context.Variables.GetValueOrDefault<string>("requestHeaders")),
                    timestamp               = new DateTimeOffset(context.Timestamp).ToUnixTimeMilliseconds(),
                    requestIpAddress        = context.Request.IpAddress,
                    statusCode              = context.Response.StatusCode,
                    responseBody            = responseBodySizeExceedsLimit ? responseBody.Substring(0, bodySizeLimit) : responseBody,
                    responseBodyTruncated   = responseBodySizeExceedsLimit,
                    responseHeaders         = context.Response.Headers,
                });
            }
        </set-body>
    </send-request>
    <!-- End of sending data to Cortex -->
```

> **IMPORTANT:**
>
> If you want to add additional data to the <outbound>, add it at the start of the <outbound> code.

3. Click Save. Your APIM traffic collection is now configured.

   Request and response data for the configured endpoints are sent to Cortex XSIAM for inspection by API security modules.

5. Go to Azure API Management data source to validate that data is ingested from Azure APIM.
6. Do the following to remove the integration of Azure APIM with Cortex XSIAM:
   - Remove the snippets you added to the policies.
   - Remove the named values from the API service.
   - Delete the HTTP log collector from Data Sources in Cortex.

1.6.7.3 | Ingest Apigee Proxy

Abstract

Integrate Cortex with Apigee Proxy to scan APIs for threats and vulnerabilities.

> **LICENSE TYPE:**
>
> Requires the Data Collection add-on.

Integrate Apigee Proxy with Cortex XSIAM to begin scanning the APIs for potential threats and vulnerabilities.

The integration uses the Apigee's JavaScript (JS) policy, implemented within a shared flow and deployed as a pre-proxy and post-proxy flow-hook in selected environments. The JS policy is designed to capture both request and response data from all traffic entering and exiting the proxy.

Settings in Cortex XSIAM

In Cortex XSIAM, set up the Apigee data source to integrate with the Apigee Gateway.

1. From Settings → Data Sources , click 

and search for Apigee and click Connect or Connect Another Instance.
2. In the Apigee Collector wizard, enter a relevant name and then click Create and Proceed.
3. Copy the key and paste it somewhere so that you can access it for later.

   If you forget to record the key and close the window, you must generate a new key and repeat this process.

4. Click the Download Configuration Script link to download the plugin, which you can then upload to the Apigee Gateway.
5. Click Close.
6. An instance is created for the data source. Click  (Copy API URL) to establish a connection when setting up Apigee Gateway.

First, download the resource file and then select the method to set up the integration with Apigee.

Run an automated script to deploy configurations to Apigee

Use the script for full deployment (with or without connecting a flow hook).

> **NOTE:**
>
> The steps include the prerequisites that run the automated script that deploys files and configurations to Apigee. For manual configuration, refer to the section Manual deployment.

1. Check that the GCP user running the script has `IAM` permissions.

   ```
   apigee.resourcefiles.list
   apigee.resourcefiles.create
   apigee.resourcefiles.update
   apigee.sharedflows.get
   apigee.sharedflows.create
   apigee.deployments.create
   apigee.sharedflowrevisions.deploy
   apigee.flowhooks.attachSharedFlow
   apigee.keyvaluemaps.create
   apigee.keyvaluemaps.delete
   apigee.keyvaluemapentries.create
   ```

2. Run the `deploy.sh` script:

   ```
   chmod +x
   ./deploy.sh
   ```

3. Verify that the JavaScript policies have been added to the shared flows:

   Go to Apigee → Proxy development → Shared Flows and check that the following policies have been added.

   - `sf-api-sec-extension-postflow`
   - `sf-api-sec-extension-preflow`

4. Validate data ingestion:

   Send a request to the gateway and go to Apigee data source to validate that the data has been received from Apigee.

5. (Optional) Exclude unwanted domains from being tracked by APIsec:

   a. Uncomment: DOMAIN_EXCLUSION_LIST.

   b. Add the domains to exclude.

   c. Edit `deploy.sh` and set the following variables:

   ```
   export DOMAIN_EXCLUSION_LIST="domain1,domain2"
   ```

6. Discontinue the integration:

   a. Edit `undeploy.sh`:

   ```
   export PROJECT_ID=example-project-id
   export ORG=$PROJECT_ID
   export ENVIRONMENT=example-env
   ```

   b. Run the undeploy.sh script:

   ```
   chmod +x
   ./undeploy.sh
   ```

   Go to Apigee → Proxy development → Shared Flows and check that the following policies have been removed.

   - `sf-api-sec-extension-postflow`
   - `sf-api-sec-extension-preflow`

Configure Apigee's JavaScript for manual deployment

You can customize the shared flow and apply it to an existing flow hook (pre-proxy, post-proxy).

Set up Apigee's JavaScript policy to send Apigee Collector's API data to Cortex XSIAM.

> **NOTE:**
>
> If you have an existing hookand would like to integrate with the shared flow, run the `deploy.sh` script, and select `n`' and exit at the prompt to create a new hook. Refer to the section Connect to existing hook.

1. Edit `panw-api-sec-extension-configuration.properties` file:

- Enter the `targetUrl` and `projectID`.
- You can update 127KB of `maxBodyInspectionSizeKB`.
- For domain exclusion, uncomment the line and add the URL to exclude.

```
targetUrl=<Cortex collector url>
projectID=<GCP project id of apigee>
maxBodyInspectionSizeKB=127 // This is default
and can be modified if needed.
commonBinaryContentType=audio/,video/,image/,
application/octet-stream,application/ogg,application/
pdf,application/zip,application/gzip,application/
vnd.rar,application/x-7z-compressed
#domainExclusionList=example.com,example2.com/shopping
```

2. Upload the edited `property set`:

   a. Get a token to upload updates via an API request. For more information, refer to property sets.

   Input:

   ```
   gcloud config config-helper --force-auth-refresh --format
   ```

   Output:

   ```
   configuration:
     active_configuration:
     properties:
       compute:
         region:
         zone:
   core:
         account:
         disable_usage_reporting:
         project:
   credential:
     access_token: <Copy this value>
     id_token:
     token_expiry:
   sentinels:
     config_sentinel:
   ```

   b. Copy the `<access_token>` value from the output.

3. Upload the `property set` to Apigee:

```
curl --silent -X GET
"https://apigee.googleapis.com/v1/organizations/
<ORG>/environments/<ENVIRONMENT>/resourcefiles/
properties" -H
"Authorization: Bearer <access_token from above>"
```

4. Generate Key Value Map (KVM), which stores the Cortex API key that's encrypted

```
curl --silent -X POST
"https://apigee.googleapis.com/v1/organizations/
<ORG>/environments/<ENVIRONMENT>/keyvaluemaps" -H
"Authorization: Bearer <access_token from above>"
-H "Content-Type: application/json" --data-raw
'{"name": "'"APISec-KVM"'", "encrypted": true}'
```

   If there's an error when creating the KVM because of an existing name, delete the KVM and recreate.

```
curl --silent -X DELETE
"https://apigee.googleapis.com/v1/organizations/
<ORG>/environments/<ENVIRONMENT>/keyvaluemaps/
$APISEC_KVM_NAME" -H "Authorization: Bearer
<access_token from above>"
```

   Add the Cortex API key entry to the created KVM.

```
curl --silent -X POST "https://apigee.googleapis.com/
v1/organizations/<ORG>/environments/<ENVIRONMENT>/
keyvaluemaps/$APISEC_KVM_NAME/entries" -H
"Authorization: Bearer <access_token from above>"
-H "Content-Type: application/json" --data-raw
'{"name": "api-key","value": "'"<Generated key
from cortex env>"'"}'
```

5. Upload the shared flows:

   Shared flows:

   - `sf-api-sec-extension-postflow`
   - `sf-api-sec-extension-preflow`

Upload

Replace the <sf> with the shared flows:

```
curl --silent -X POST --data-binary "<sf>.zip" -H
"Content-Type: application/octet-stream" -H
"Authorization: Bearer <access_token from above>"
"https://apigee.googleapis.com/v1/organizations/$ORG/
sharedflows?action=import&name=<sf>"
```

Deploy

Input:

```
curl --silent -X GET "https://apigee.googleapis.com/
v1/organizations/<ORG>/sharedflows/<sf>" -H
"Authorization: Bearer <access_token from above>"
```

Output:

```
{
  "metaData": {
    "createdAt": "1736952161610",
    "lastModifiedAt": "1736952161610",
    "subType": "SharedFlow"
  },
  "name": "sf-api-sec-extension-postflow",
  "revision": [
    "1" // This is the revision number
  ],
  "latestRevisionId": "1"
}
```

6. Deploy <sf>:

```
curl --silent -X POST -H "Authorization:
Bearer <access_token from above>"
"https://apigee.googleapis.com/
v1/organizations/$ORG/environments/<ENVIRONMENT>/
sharedflows/$sf/revisions/<REVISION>/
deployments?override=true"
```

7. Verify API security shared flows were created:

   Go to Apigee → Proxy development → Shared Flows and check that the following policies have been added.

   - `sf-api-sec-extension-postflow`
   - `sf-api-sec-extension-preflow`

Connect to an existing hook

Follow the steps if you have an existing hook and would like to integrate with a shared flow.

1. Check for existing flow hooks.

   a. Go to Apigee → Management → Environments and select the environment to hook the shared flow.

   b. In the Flow Hooks tab, select the relevant flow hook.

2. Configure policy for shared flow to the existing hook.

   a. Go to Apigee → Proxy development → Shared Flows and select the flow hook from the relevant environment.

   > **NOTE:**
   >
   > Start with the hook in pre-proxy.

   b. From the Develop tab, expand Policies and select Flow Callout.

   c. Enter a meaningful name and select the Sharedflow: `sf-api-sec-extension-preflow` , and then click Create.

   d. From the Develop tab, select Shared flows and expand Default.

   e. From Select policy, select Select existing policy, and select the policy just created and then click Add.

   f. Repeat the previous steps for the post-proxy hook. Select the Sharedflow: `sf-api-sec-extension-postflow`.

   g. Click Save and Deploy.

   The steps automatically run without linking to the hooks.

   > **IMPORTANT:**

> This should only be done when there are already existing hooks, and API security shared flows can't be hooked as a standalone. Run the deployment script, but skip step 9 by passing n. This step publishes API security shared flows to the desired Apigee environment without setting them to flow hooks.

3. Limitations:

   - The API security extension deployment scripts currently do not support archive-deployment Apigee environments. Refer to Manage archive deployment for more information.

     > **NOTE:**
     >
     > Archive deployments are currently in preview and are subject to change.

   - The API security extension for Apigee relies on flow-hooks, which are available only with Intermediate or Comprehensive Apigee Environment types. Refer to Environments for more information.
   - For requests/responses with binary payloads, the binary payload is not sent to the collector for analysis; only the metadata (for example, HTTP headers, query parameters, etc.) is sent.

1.6.7.4 | Ingest Kong

Abstract

Integrate Cortex with Kong to scan APIs for threats and vulnerabilities.

> **LICENSE TYPE:**
>
> Requires the Data Collection add-on.

Integrate Kong with Cortex XSIAM to start scanning its APIs for potential threats and vulnerabilities.

You need to integrate a dedicated Kong HTTP log plugin. This plugin enables seamless traffic ingestion from your Kong API gateway to Cortex XSIAM, allowing for comprehensive security measures such as OWASP Top-10, bot detection, access control, and more.

Settings in Cortex XSIAM

In Cortex XSIAM, set up the Kong data source to integrate with the Kong API Gateway.

1. From Settings → Data Sources , click 

and search for Kong , and then click Connect or Connect Another Instance.
2. In the Kong Collector wizard, enter a relevant name and then click Create and Proceed.
3. Copy the key and paste it somewhere so that you can access it later.

   If you forget to record the key and close the window, you must generate a new key and repeat this process.
4. Click the Download Custom Plugin link to download the plugin, which you can then install on the Kong API Gateway.
5. Click Close.
6. An instance is created for the data source. Click  (Copy API URL) to establish a connection when setting up Kong API Gateway.

Follow the steps to integrate Kong's API gateway with Cortex XSIAM.

Provision Kong API gateway with the custom plugin

To deploy the custom plugin, refer to the Kong API documentation online:

- Kong Gateway
- Kong Konnect
- Kong Ingress Controller

Example 162. Kong as docker container

In this example, we'll use Docker to deploy Kong with the custom plugin.

1. Extract the zip archive into some target directory TARGET_DIRECTORY. This directory should contain an extracted "kong" directory.
2. Add the following arguments to the `docker run` command, replacing TARGET_DIRECTORY with the correct path:

```
-v "TARGET_DIRECTORY/kong:/tmp/custom_plugins/kong" \
-e "KONG_LUA_PACKAGE_PATH=/tmp/custom_plugins/?.lua;;" \
-e "KONG_PLUGINS=bundled,panw-apisec-http-log"
```

You may want to adjust the size of the nginx body buffer, which is used by Kong internally. This size sets the upper limit on the amount of HTTP body bytes that can be mirrored by the plugin. By default, this value is 8192 bytes (8 KB). To change it, another argument can be passed to the docker: for example, setting it to 128 KB:

```
-e "KONG_NGINX_HTTP_CLIENT_BODY_BUFFER_SIZE=128k"
```

See https://nginx.org/en/docs/syntax.html for information on the allowed values of this variable.

> **IMPORTANT:**
>
> The size of the buffer must be equal to or larger than the max body size setting in the plugin configuration, on every data plane node.

3. To verify that the plugin is installed, query Kong's Admin API using the following command:

```
curl admin-api-hostname:8001 | jq .configuration.loaded_plugins.'"panw-apisec-http-log"'
```

This prints true to the terminal if the plugin is loaded into the Kong instance.

Add and configure the custom plugin

Add and configure the plugin.

1. From the Kong Manager menu, go to Plugins.
2. From the Plugins page, scroll down to the Custom Plugins section.
3. Select panw-apisec-http-log and click Edit to configure the panw-apisec-http-log plugin settings.

Configuration	Description	Example
Protocols	The request protocols the plugin will be applied to.	Either http, https, or both
Cloud Context	Cloud context, such as AWS Account ID, GCP Project ID, Azure Subscription or an appropriate value for on-prem.	987654321000
Cloud Provider	Cloud provider where Kong API Gateway is installed.	AWS.
Cloud Region	Cloud region.	us-east-2
Cloud API Key	The collector authorization key provided by the Cortex platform.	
HTTP Endpoint	The Cortex collector's endpoint URL.	

4. Click the View Advanced Parameters to configure optional settings.

> **NOTE:**
>
> The queue parameters can be updated to change when the plugin mirrors data to Cortex.

Configuration	Description	Example
Instance Name	A custom name for this plugin instance. This is useful when applying different instances to different scopes.	Empty
Tags	An optional set of strings for grouping and filtering, **NOTE:** Use commas to separate tags.	Empty

Configuration	Description	Example
Keepalive	An optional value in milliseconds that defines how long an idle connection will live before being closed.	60000 (60 seconds)
Timeout	An optional timeout in milliseconds when sending data to Cortex.	10000 (10 seconds)
Max body size	The maximum body size to mirror in bytes (for example: 1024 is 1KB). Any bytes beyond this size are omitted from the request and response bodies. Must be <= 4 MB and <= the value of Kong's nginx_http_client_body_buffer_size setting.	131072 (128 KB), or the nginx body buffer size if it's smaller.
Queue Concurrency Limit	The number of queue delivery timers. -1 indicates unlimited.	1
Queue.Initial Retry Delay	Time in seconds before the initial retry is made for a failing batch.	0.01 (10 milliseconds)
Queue.Max Batch Size	Maximum number of entries that can be processed at a time.	1
Queue.Max Bytes	Maximum number of bytes that can be waiting in a queue, requires string content	Unlimited
Queue.Max Coalescing Delay	Maximum number of (fractional) seconds to elapse after the first entry was queued before the queue starts calling the handler.	1
Queue.Max Entries	Maximum number of entries that can be waiting in the queue.	10000
Queue.Max Retry Delay	Maximum time in seconds between retries, caps exponential backoff.	60
Queue.Max Retry Time	Time in seconds before the queue gives up calling a failed handler for a batch.	60

5. Go to Kong data source to validate that data is ingested from the Kong API Gateway.

Limitations

- The plugin supports HTTP and HTTP/S protocols.
- The plugin supports Kong API Gateway version 3.4.x and above.
- The nginx body buffer size on each data plane node must be equal or larger than the max body size setting.
- Request and response bodies will not be mirrored if their size exceeds the nginx body buffer size. When this occurs, it is indicated in the metadata that is sent to Cortex along with the HTTP transaction data.
- The mirrored response body is the body returned from the upstream service. This means that changes made to the response body by other plugins, is not reflected in the mirrored data.

1.6.8 | Ingest data from third-party pipeline solutions

Abstract

Ingest data from third-party pipeline solutions into Cortex XSIAM.

Ingest data from third-party pipeline solutions into Cortex XSIAM.

1.6.8.1 | Ingest data from Cribl

Abstract

Ingest third-party data collected by Cribl.

> **NOTE:**
>
> The Cribl data collector is a beta feature.

The Cribl data collector is an out-of-the-box native integration which ingests data that Cribl collects from multiple data sources and streams to Cortex XSIAM, while ensuring that all downstream capabilities, including analytics, are available in Cortex XSIAM.

The onboarding process in Cribl has an impact on the output that is sent to Cortex XSIAM. Therefore, the onboarding process of some sources in Cribl might have to be implemented in a certain way in order to adhere to Cortex XSIAM requirements. These processes are described in more detail in Tasks 1 and 3, below.

Raw data must be collected by Cribl and streamed as-is from the passed through source, because any changes made by Cribl might affect the way that Cortex XSIAM handles the data.

For best results, we recommend ingesting data from Palo Alto Networks products, such as Next-Generation Firewall (NGFW) using the dedicated Cortex XSIAM data collectors, instead of source collectors provided by Cribl. Although you can ingest FW data through Cribl, ingesting it that way will omit a layer of data (EAL).

> **NOTE:**
>
> We do not support email data collection via Cribl.

Workflow high-level overview:

1. Task 1: In Cribl, onboard data collection from your data sources.
2. Task 2: In Cortex XSIAM, create a Cribl data collector instance, and obtain the authorization token and the API URL.
3. Task 3: In Cribl, for each source, configure the destination, using the Cortex XSIAM authorization token, the Cortex XSIAM API URL, and the source UUID.
4. Task 4: Verify that data is streamed to Cortex XSIAM as expected, and perform ongoing maintenance.

Perform the following tasks in the order that they appear.

Task 1 (in Cribl, create new data sources)

> **PREREQUISITE:**
>
> Ensure that you have the credentials and IDs for each data source, such as Tenant ID, App ID and Client secret.

General guidelines specifically for Cortex XSIAM (for more information, refer to Cribl documentation):

- If you have not already done so, create source collectors to onboard the desired data sources.
- For some data sources, Cribl includes specific collectors in its catalog. If you can't find one in the catalog specifically for your source, use a generic collector.
- Although some native Cribl source collectors allow you to ingest several data types using the same source collector, we do not recommend this approach. To ensure optimal Cortex XSIAM performance, configure a separate Cribl source collector for each data type. For reference purposes, this data source UUID list shows all the data types that can be onboarded.

  For example, Microsoft 365 has several data types, such as users, groups, and contacts.

Task 2 (in Cortex XSIAM)

> **NOTE:**
>
> Only one Cribl data collector instance can be configured in Cortex XSIAM.

1. Go to Settings → Configuration → Data Collection → Data Sources.
2. Search for Cribl.
3. Click the Cribl integration, and then click Connect.
4. In the Name field, enter a meaningful name for the integration.
5. Click Save & generate token.
6. Click the Copy icon to copy the authorization token.

7. Save the authorization token copy in a safe place for future use. You cannot access this token again, so take care to copy it and save it before you close the dialog box.

8. Click Close.

9. On the Data Sources page, in the row for the Cribl instance, click the link icon (Copy API URL). Save the API URL copy in a safe place for future use.

Task 3 (In Cribl, configure Cortex XSIAM as a destination for each source)

> **PREREQUISITE:**
>
> Ensure that you have the copies of the Cortex XSIAM authorization token and API URL obtained in Task 2.

The following table includes guidelines that are relevant specifically for Cortex XSIAM. While you are configuring Cortex XSIAM destinations for your sources, configure the items listed in the table below as described.

For general information about configuring destinations, refer to Cribl documentation.

Item	Setting	Details
Cortex XSIAM URL	XSIAM Endpoint field	Paste the API URL obtained from Cortex XSIAM.
Authorization token	Authorization Token field	Paste the authorization token obtained from Cortex XSIAM.
Advanced Settings	Compress toggle	Ensure that Compress is disabled.
HTTP headers	Extra HTTP headers	Add extra HTTP headers for each data source: • Source-identifier: Search the table supplied in this topic for the vendor and product. The Data source UUIDs table lists the data sources that can be identified by Cortex XSIAM, using their corresponding UUIDs. These UUIDs are required to map data collected by Cribl to the Cortex XSIAM destination. ◦ If you find the desired vendor and product, copy the corresponding UUID from the table and paste here. This UUID will allow Cortex XSIAM to leverage all the data ingested from the data source, such as identifiers, pipeline sources such as IP addresses, devices, and so on. Data from sources known to Cortex XSIAM are saved in the appropriate datasets–the same datasets as those used by dedicated data collectors in Cortex XSIAM. Note: Do not use the generic UUID for a data source that is known to Cortex XSIAM and appears in the table. ◦ If you can't find the desired vendor and product source in the table, copy the generic UUID provided in the first row of the table, and paste here. Data from unknown sources are saved in a separate searchable dataset. The dataset name will reflect the Vendor and Product names that you enter next. • Format: json • Vendor: When using the UUID for unknown data sources, you must enter the vendor name. • Product: When using the UUID for unknown data sources, you must enter the product name.
Mapping	Passthru option	Map the data source(s) that you created in Task 1 to the XSIAM data destination created in this task. Ensure that you select the Passthru option.
Deployment	Commit & Deploy Deploy	When mapping is complete, click Commit & Deploy, and then click Deploy.

Task 4 (in Cribl and in Cortex XSIAM)

Verify that data is streaming as expected from Cribl to Cortex XSIAM.

- In the Cribl user interface, click the Source collector, click Configure, and then the Charts tab. Check the charts to verify that streaming is in progress.
- In Cortex XSIAM, on the Data Sources page, when streaming begins, a green check mark appears below the Cribl configuration, along with the amount of data received.

Other optional tasks

Use the Disable and Delete options with extreme caution.

- Disabling the integration will cease streaming from Cribl.
- Deleting the integration will erase the integration completely and will require reconfiguration, because the original authorization token will be lost.

Disable the integration with Cribl

1. To disable the integration, in Cortex XSIAM, search for the Cribl integration on the Data Sources page, and clear the Enable checkbox.
2. In the Are you sure? dialog box, type `disable`, and then click Disable.

Delete the integration with Cribl

1. To delete the Cribl integration, in Cortex XSIAM, search for the integration on the Data Sources page, and click the integration's Delete icon.
2. In the Are you sure? dialog box, type `delete`, and then click Delete.

1.6.8.1.1 | Data source UUIDs

Abstract

Data source UUIDs

Vendor	Product	UUID
		af01292940d7426594d3d3e55ae17ee0 **NOTE:** Do not use this generic UUID when your data source is listed in this table.
Salesforce	Salesforce logs	ab109687acd24978aabcb7ad8b5742e3
Salesforce	Salesforce snapshots	addbf31a6372491e88d45934dff5b5b0
Dropbox	Events	a6322b2fd9e545e0a4223ba754c48fb9
Dropbox	Directory	e8d2c52bc9594621924fab0507264586
Workday	Workday	00d4e740702d4eb2939a87c2318513dd
Google	Cloud Logging	00a8322c85e14beabfa7ad5f3d62db73
Okta	SSO	5faf4c1fdb8443d9920d6a54815432c1
Microsoft	Azure AD	c00d6d52e5b141a8baa8db9d9345423d
PingID	PingONE	924951a8394b4605b1725f943292ab4f
CrowdStrike	Falcon incident	230b2b0233bf4327806af72e6e5769f3

Vendor	Product	UUID
CrowdStrike	Hosts	8b673ac8e2f34b4a8dc14c22f0e6063b
CrowdStrike	Falcon Data Replicator	6cd7d60f0ff5497baecf6b9073c8000e
SentinelOne	Deep Visibility	b9fa55e6fa564c709358425ce0f61517
Microsoft	Azure	fce13a1d51294f84bae4a37851503060
Microsoft	Defender	ce9e8cf36e0742c38aa89787a256855f
ServiceNow	CMDB	8b3e767247e44471a95e563378d0b9be
Prisma	Cloud	f8c3403a02fd4147862ee293bf4e74e2
Prisma	Assets	6a61c1cba1b64cd2a977c76c41f7950d
Proofpoint	TAP	3eefce0f791e4391a3643b8cf860a361
Microsoft	Office 365 Exchange Online	dee8e85ce7db4573a8bc21b807e1d73a
Microsoft	Azure AD audit	0e076d5abe864bf78e8145ea9e0d749e
Microsoft	Azure AD sign-ins	f56dcfdf6bca43e793a4b6e9290e7b12
Microsoft	Graph security alerts	5619f2f691fc46c4b202587fdaa031c3
Palo Alto Networks	IOT Security devices	80cee50bfc6e4ac5b34b19794b767acd
Palo Alto Networks	IOT Security alerts	e772949c88ec4107ad81ec38061d35c0
Google	Gsuite reports	3ddd43030db142839568943e0a2fe785
Google	Gmail	8607490288d1407ba82b5c5ad9dc64a0
Box	Box	3ef05d14ae9349f8bbd48c8a4797334a
OneLogin	Events	22b23a3f9f1e49998645b683d5dc3a6f
OneLogin	OneLogin	88cfbd3e7b974d999b10edac83995b8a
Google	Workspace alerts	4f263650cd29475c81f2ff953cf19827

Vendor	Product	UUID
Microsoft	Office 365 devices	de229685f708413fad46289657ea09de
Microsoft	Office 365 rules	6b925df8923d4038bf78998d1ffde77c
Microsoft	Office 365 users	dcfb7a412e654efd868de0b8cf81766a
Microsoft	Office 365 groups	0b0499ac0d984145b201c6d674771dbf
Microsoft	Office 365 contacts	de1b694a6c8341958bc08c4b7c140874
Microsoft	Office 365 domains	cae29fd87b554bd9a5694afb225e8dc9
Microsoft	Office 365 mailboxes	9855a03559ce4263b568671e695d1fa8
Google	Workspace ChromeOS devices	e82ae276e6b9442fa80920a03d2a38d6
Google	Workspace groups	689ae8ef14e848e3855b81e91d8af9bc
Google	Workspace domains	2738e963ac3141afaad05885e060a73c
Google	Workspace profiles	f2aed57ff13c439eb93153ba7309fe87
Google	Workspace rules	2621aaf3334a4147ae727afe84db31a9
Google	Workspace organization	4e342367057d46c7b38ce7d40682fd1e
Google	Workspace group members	8a0140fc47b643838d0fcf096773c0a1
Google	Workspace contacts	d20d6cfea3e943d5a5a6bc005c429ef4
Google	Workspace users	359ecd845fa54caab6ddb4b7c7a2764d
Google	Workspace mailboxes	328796d692f343c38f07351e8c783f80
Google	Workspace users send as aliases	3b8f9e65f8ed43f4a4e5679236691fe2
Google	Workspace schemas	c978986a6b3846c7b6fdbe15bef14f69
Google	Workspace mailbox settings	3c4beffadfac40a18aaf4d143a19dc27

Vendor	Product	UUID
Google	Workspace privileges	462ebcbfce9341ac8c006e5aa45ccf44
Google	Workspace mobile devices	149b58ec938d4d1a8568359483e50800
Google	Workspace roles	82170b42b9684b79bda124c712bcbdc0
Google	Workspace contact groups	5a42004787064021a462bb2120160514
Google	Workspace apps	5a617df8827d461db66a10d084c7b39f
Amazon	AWS EKS	fb8a9d4922cb4095b76d71e921d2d999
Microsoft	DHCP	b55819e8959c49728d5d98a6d87eafb6
Amazon	AWS flow logs	667083aa68544eee8b67cdd2d4cc327b
Amazon	AWS audit logs	c19f87b6262f48259b3d5d2a2c691802
Amazon	AWS generic logs	0498f8a24de04b3e85102e742f6783f8
Amazon	AWS Route 53 logs	d57ae82c1e2a4d138fc34084d159b09e
Amazon	AWS prompt logs	a53edad7ef0c46ffb5037fb2e21520cb
Microsoft	Office 365 Sharepoint Online	3a37f519e9094a3f8c4185fa572cd111
Microsoft	Office 365 Azure AD	e1f109f886ea42fbb96be6ec0cc597a9
Microsoft	Office 365 DLP	8f052782739d4b8389644cca23b994ac
Microsoft	Office 365 General	c7655e83805b4a058e66043a6715156c

1.6.9 | Additional log ingestion methods

Abstract

Cortex XSIAM supports custom log ingestion methods.

In addition to native log ingestion support, Cortex XSIAM also supports a number of custom log ingestion methods.

1.6.9.1 | Ingest logs from a Syslog receiver

Abstract

To extend visibility, Cortex XSIAM can receive Syslog from additional vendors that use CEF or LEEF formatted over Syslog (TLS not supported).

Cortex XSIAM can receive Syslog from a variety of supported vendors (see External data ingestion vendor support). In addition, Cortex XSIAM can receive Syslog from additional vendors that use CEF, LEEF, CISCO, CORELIGHT, or RAW formatted over Syslog.

After Cortex XSIAM begins receiving logs from the third-party source, Cortex XSIAM automatically parses the logs in CEF, LEEF, CISCO, CORELIGHT, or RAW format and creates a dataset with the name `<vendor>_<product>_raw`. You can then use XQL Search queries to view logs and create new IOC, BIOC, and Correlation Rules.

To receive Syslog from an external source:

1. Set up your Syslog receiver to forward logs.
2. Activate the Syslog collector applet on a Broker VM within your network. For more information, see Activate the Syslog Collector.
3. Use the XQL Search to search your logs.

1.6.9.2 | Ingest Apache Kafka events as datasets

Abstract

Cortex XSIAM can receive logs and data from Apache Kafka directly to your log repository for query and visualization purposes.

Cortex XSIAM can receive events from Apache Kafka clusters directly to your log repository for query and visualization purposes. After you activate the Kafka Collector applet on a Broker VM in your network, which includes defining the connection details and settings related to the list of subscribed topics to monitor and upload to Cortex XSIAM, you can collect events as datasets.

After Cortex XSIAM begins receiving topic events from the Kafka clusters, Cortex XSIAM automatically parses the events and creates a dataset with the specific name you set as the target dataset when you configured the Kafka Collector, and adds the data in these files to the dataset. You can then use XQL Search queries to view events and create new Correlation Rules.

Configure Cortex XSIAM to receive events as datasets from topics in Kafka clusters.

1. Activate the Kafka collector applet on a Broker VM within your network. For more information, see Activate the Kafka Collector.
2. Use the XQL Search to query and review logs.

1.6.9.3 | Ingest CSV files as datasets

Abstract

Cortex XSIAM can receive CSV log files from a shared Windows directory, where the CSV log files must conform to specific guidelines.

Cortex XSIAM can receive CSV log files from a shared Windows directory directly to your log repository for query and visualization purposes. After you activate the CSV Collector applet on a Broker VM in your network, which includes defining the list of folders mounted to the Broker VM and setting the list of CSV files to monitor and upload to Cortex XSIAM (using a username and password), you can ingest CSV files as datasets.

The ingested CSV log files must conform to the following guidelines:

- Header field names must contain only letters (a-z, A-Z) or numbers (0-9) and must start with a letter. Spaces are converted to underscores (_).
- Date values can be in either of the following formats:
    - YYYY-MM-DD (optionally including HH:MM:SS)
    - Unix Epoch time. For example, 1614858795.

After Cortex XSIAM begins receiving logs from the shared Windows directory, Cortex XSIAM automatically parses the logs and creates a dataset with the specific name you set as the target dataset when you configured the CSV Collector. The CSV Collector checks for any changes in the configured CSV files, as well as any new CSV files added to the configuration folders, in the Windows directory every 10 minutes and replaces the data in the dataset with the data from those files. You can then use XQL Search queries to view logs and create new Correlation Rules.

Configure Cortex XSIAM to receive CSV files as datasets from a shared Windows directory.

1. Ensure that you configure Windows to share the applicable CSV files in your Windows directory.
2. Activate the CSV collector applet on a Broker VM within your network. For more information, see Activate the CSV Collector.
3. Use the XQL Search to locate and review logs.

1.6.9.4 | Ingest database data as datasets

Abstract

Cortex XSIAM can receive data from a client relational database directly to your log repository.

Cortex XSIAM can receive data from a client relational database directly to your log repository for query and visualization purposes. After you activate the Database Collector applet on a Broker VM in your network, which includes defining the database connection details and settings related to the query details for collecting the data from the database to monitor and upload to Cortex XSIAM, you can collect data as datasets. For more information about activating this collector applet, see Activate the Database Collector.

After Cortex XSIAM begins receiving data from a client relational database, Cortex XSIAM automatically parses the logs and creates a dataset with the specific name you set as the target dataset when you configured the Database Collector using the format `<Vendor>_<Product>_raw`. The Database Collector checks for any changes in the configured database based on the SQL Query defined in the database connection according to the execution frequency of collection that you configured and appends the data to the dataset. You can then use XQL Search queries to view data and create new Correlation Rules.

Configure Cortex XSIAM to receive data as datasets data from a client relational database.

1. Activate the Database Collector applet on a Broker VM within your network.
2. Use the XQL Search to query and review logs.

1.6.9.5 | Ingest logs in a network share as datasets

Abstract

Cortex XSIAM can receive logs from files and folders in a network share directly to your log repository for query and visualization purposes.

Cortex XSIAM can receive logs from files and folders in a network share directly to your log repository for query and visualization purposes. After you activate the Files and Folders Collector applet on a Broker VM in your network, which includes defining the connection details and settings related to the list of files to monitor and upload to Cortex XSIAM, you can collect files as datasets.

After Cortex XSIAM begins receiving logs from files and folders in a network share, Cortex XSIAM automatically parses the logs and creates a dataset with the specific name you set as the target dataset when you configured the Files and Folders Collector using the format `<Vendor>_<Product>_raw`. The Files and Folders Collector reads and processes the configured files one by one, as well as any new files added to the configured files and folders, in the network share according to the execution frequency of collection that you configured and adds the data in these files to the dataset. You can then use XQL Search queries to view logs and create new Correlation Rules.

> **NOTE:**
>
> The Files and Folders Collector applet only starts to collect files that are more than 256 bytes.

Configure Cortex XSIAM to receive logs as datasets from files and folders in a network share.

1. Activate the Files and Folders collector applet on a Broker VM within your network. For more information, see Activate the Files and Folders Collector.
2. Use the XQL Search to query and review logs.

1.6.9.6 | Ingest FTP files as datasets

Abstract

Cortex XSIAM can receive logs from files and folders via FTP, FTPS, and SFTP directly to your log repository for query and visualization purposes.

Cortex XSIAM can receive logs from files and folders via FTP, FTPS, or SFTP directly to your log repository for query and visualization purposes. After you activate the FTP Collector applet on a Broker VM in your network, which includes defining the connection details and settings related to the list of files to monitor and upload to Cortex XSIAM, you can collect files as datasets.

After Cortex XSIAM begins receiving logs from files and folders via FTP, FTPS, or SFTP, Cortex XSIAM automatically parses the logs and creates a dataset with the specific name you set as the target dataset when you configured the FTP Collector using the format `<Vendor>_<Product>_raw`. The FTP Collector reads and processes the configured FTP files one by one, as well as any new FTP files added to the configured files and folders, in the FTP directory according to the execution frequency of collection that you configured, and adds the data in these files to the dataset. You can then use XQL Search queries to view logs and create new Correlation Rules.

Configure Cortex XSIAM to receive logs as datasets from files and folders via FTP, FTPS, or SFTP.

1. Activate the FTP collector applet on a Broker VM within your network. For more information, see Activate the FTP Collector.
2. Use the XQL Search to query and review logs.

1.6.9.7 | Ingest NetFlow flow records as datasets

Abstract

Cortex XSIAM can receive NetFlow flow records and IPFIX from a UDP port directly to your log repository for query and visualization purposes.

Cortex XSIAM can receive NetFlow flow records and IPFIX from a UDP port directly to your log repository for query and visualization purposes. After you activate the NetFlow Collector applet on a Broker VM in your network, which includes configuring your NetFlow Collector settings, you can ingest NetFlow flow records and IPFIX as datasets.

The ingested NetFlow flow record format must include, at the very least:

- Source and Destination IP addresses
- TCP/UDP source and destination port numbers

When Cortex XSIAM begins receiving flow records from the UDP port, Cortex XSIAM automatically parses the flow records and creates a dataset with the specific name you set as the target dataset when you configured the NetFlow Collector. The NetFlow Collector adds the flow records to the dataset. You can then use XQL Search queries to view those flow records and create new IOC, BIOC, and Correlation Rules. Cortex XSIAM can also analyze your logs to generate Analytics issues.

Configure Cortex XSIAM to receive NetFlow flow records as datasets from the routers and switches that support NetFlow.

1. Set up your NetFlow exporter to forward flow records to the IP address of the Broker VM that runs the NetFlow collector applet.
2. Activate the NetFlow collector applet on a Broker VM within your network. For more information, see Activate the NetFlow Collector.
3. Use the XQL Search to query your flow records, using your designated dataset.

1.6.9.8 | Set up an HTTP log collector to receive logs

Abstract

You can set up Cortex XSIAM to receive logs from third-party sources, and automatically parse and process these logs.

In addition to logs from supported vendors, you can set up a custom HTTP log collector to receive logs in Raw, JSON, CEF, or LEEF format. The HTTP Log Collector can ingest up to 80,000 events per sec.

When Cortex XSIAM begins receiving logs from the third-party source, Cortex XSIAM automatically parses the logs and creates a dataset with the name `<Vendor>_< Product>_raw`. You can then use XQL Search queries to view logs and create new Correlation rules.

To set up an HTTP log collector to receive logs from an external source.

1. Create an HTTP Log collector in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select HTTP, and click Connect.

   c. Specify a descriptive Name for your HTTP log collection configuration.

   d. Select the data object Compression, either gzip or uncompressed.

   e. Select the Log Format as Raw, JSON, CEF, or LEEF.

      Cortex XSIAM supports logs in single line format or multiline format. For a JSON format, multiline logs are collected automatically when the Log Format is configured as JSON. When configuring a Raw format, you must also define the Multiline Parsing Regex as explained below.

      > **NOTE:**
      >
      > -The Vendor and Product defaults to Auto-Detect when the Log Format is set to CEF or LEEF.
      >
      > -For a Log Format set to CEF or LEEF, Cortex XSIAM reads events row by row to look for the Vendor and Product configured in the logs. When the values are populated in the event log row, Cortex XSIAM uses these values even if you specified a value in the Vendor and Product fields in the HTTP collector settings. However, when the values are blank in the event log row, Cortex XSIAM uses the Vendor and Product that you specified in the HTTP collector settings. If you did not specify a Vendor or Product in the HTTP collector settings, and the values are blank in the event log row, the values for both fields are set to unknown.

   f. Specify the Vendor and Product for the type of logs you are ingesting.

   g. (Optional) Specify the Multiline Parsing Regex for logs with multilines.

      This option is only displayed when the Log Format is set to Raw, so you can set the regular expression that identifies when the multiline event starts in logs with multilines. It is assumed that when a new event begins, the previous one has ended.

   h. Save & Generate Token.

      Click the copy icon next to the key and record it somewhere safe. You will need to provide this key when you configure your HTTP POST request and define the api_key. If you forget to record the key and close the window you will need to generate a new key and repeat this process.

      Click Done when finished.

2. Send data to your Cortex XSIAM HTTP log collector.

   a. Send an HTTP POST request to the URL for your HTTP Log Collector.

   You can view a sample curl or python request on an HTTP collector instance by selecting ⋮ View Example.

   Here is a CURL example:

```
curl -X POST https://api-{tenant external URL}/logs/v1/event -H 'Authorization: {api_key}' -H 'Content-Type: text/plain' -d '{"example1": "test", "timestamp":
1609100113039}
{"example2": [12321,546456,45687,1]}'
```

   Python 3 example:

```
import requests
def test_http_collector(api_key):
    headers = {
        "Authorization": api_key,
        "Content-Type": "text/plain"
    }
    # Note: the logs must be separated by a new line
    body = "{'example1': 'test', 'timestamp': 1609100113039}" \
           "{'example2': [12321,546456,45687,1]}"
    res = requests.post(url="https://api-{tenant external URL}/logs/v1/event",
                        headers=headers,
                        data=body)
    return res
```

   b. Substitute the values specific to your configuration.

   - `url`: You can copy the URL for your HTTP log collector from the Custom Collectors page. For example: `https://api-{tenant external URL}/logs/v1/event`.
   - `Authorization`: Paste the `api_key` you previously recorded for your HTTP log collector, which is defined in the header.
   - `Content-Type`: Depending on the data object format you selected during setup, this will be `application/json` for JSON format or `text/plain` for Text format. This is defined as part of the header.
   - `Body`: The body contains the records you want to send to Cortex XSIAM. Separate records with a `\n` (new line) delimiter. The request body can contain up to 10 Mib records, but 1 Mib is recommended. In the case of a curl command, the records are contained in the `-d ‘<records>’` parameter.

     > **NOTE:**
     >
     > Each record cannot exceed 5 MB in size.

   c. Review the possible success and failure code responses to your HTTP Post requests.

   The following table provides the various success and failure code responses to your HTTP Post requests, which can help you troubleshoot any problems with your HTTP Collector configuration.

Success/Failure Response Code	Description	Output Code Displayed (If Applicable)
200	Success code that indicates there are no errors and the request was successful.	`{ "error": "false"}`
401	Unauthorized error code that indicates either an incorrect authorization token is being used or that the HTTP Collector is deleted/disabled.	
404	Error code 404 page not found that indicates a wrong URL.	
413	Error code indicating the payload is too large as the request size limit is 10 MB.	
500	Error code indicating the request was not able to be processed due to an incorrect log format between the request and the HTTP collector configuration.	`{ "error": "error processing request, error: failed to process the request"}`

Success/Failure Response Code	Description	Output Code Displayed (If Applicable)
429	Error code indicating too many requests as the rate limit is 400 requests per second per customer per endpoint.	

3. Monitor your HTTP Log Collection integration.

   You can return to the Settings → Data Sources page to monitor the status of your HTTP Log Collection configuration. For each instance, Cortex XSIAM displays the number of logs received in the last hour, day, and week. You can also use the Data Ingestion Dashboard to view general statistics about your data ingestion configurations.

4. After Cortex XSIAM begins receiving logs, use the XQL Search to search your logs.

1.6.9.9 | Ingest logs from BeyondTrust Privilege Management Cloud

Abstract

Extend Cortex XSIAM visibility into logs from BeyondTrust Privilege Management Cloud.

If you use BeyondTrust Privilege Management Cloud, you can take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your logs to Cortex XSIAM. This enables Cortex XSIAM to help you expand visibility into computer, activity, and authorization requests in the organization, correlate and detect access violations, and query BeyondTrust Endpoint Privilege Management logs using XQL Search.

When Cortex XSIAM starts to receive logs, Cortex XSIAM can analyze your logs in XQL Search and you can create new Correlation Rules.

To integrate your logs, you first need to configure SIEM settings and an AWS S3 Bucket according to the specific requirements provided by BeyondTrust. You can then configure data collection in Cortex XSIAM by configuring an Amazon S3 data collector for a generic log type using the Beyondtrust Cloud ECS log format.

Before you begin configuring data collection verify that you are using BeyondTrust Privilege Management Cloud version 21.6.339 or later.

Configure BeyondTrust Privilege Management Cloud collection in Cortex XSIAM.

1. Configure SIEM settings and an AWS S3 Bucket according to the requirements provided in the BeyondTrust documentation.

   Ensure that when you add the AWS S3 bucket in the PMC and set the SIEM settings, you select ECS - Elastic Common Schema as the SIEM Format.

2. Configure BeyondTrust logs collection with Cortex XSIAM using an Amazon S3 data collector for generic data.

   Ensure your Amazon S3 data collector is configured with the following settings.

   - Log Type: Select Generic to configure your log collection to receive generic logs from Amazon S3.
   - Log Format: Select the log format type as Beyondtrust Cloud ECS.

     > **NOTE:**
     >
     > For a Log Format set to Beyondtrust Cloud ECS, the following fields are automatically set and not configurable.
     >
     > - Vendor: Beyondtrust
     > - Product: Privilege Management
     > - Compression: Uncompressed

3. After Cortex XSIAM begins receiving data from BeyondTrust Privilege Management Cloud, you can use XQL Search to search your logs using the `beyondtrust_privilege_management_raw` dataset that you configured when setting up your Amazon S3 data collector.

1.6.9.10 | Ingest logs and data from Box

Abstract

Ingest logs and data from Box enterprise accounts via the Box REST APIs.

Cortex XSIAM can ingest different types of data from Box enterprise accounts using the Box data collector. To receive logs and data from Box enterprise accounts via the Box REST APIs, you must configure the Data Sources settings in Cortex XSIAM based on your Box enterprise account credentials. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset for the different types of data that you are collecting, which you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library. For all logs, Cortex XSIAM can generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC), when relevant, from Box logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

The following table provides a brief description of the different types of data you can collect, the collection method and fetch interval for new data collected, the name of the dataset to use in Cortex XSIAM to query the data using XQL Search, and whether the data is normalized.

NOTE:

The Fetch Intervals are non-configurable.

Type Of Data	Description	Collection Method	Fetch Interval	Dataset Name	Normalized Data
Events and security alerts					
Events (admin_logs)	Retrieves events related to file/folder management, permission changes, access and login activities, user/groups management, folder collaboration, file/folder sharing, security settings changes, tasks, permission changes on folders, storage expiration and data retention, and workflows.	Appends data	60 seconds	`box_admin_logs_raw`	When relevant, Cortex XSIAM normalizes SaaS audit event logs into stories, which are collected in a dataset called `saas_audit_logs`.
Box Shield Alerts	Retrieves security alerts related to suspicious locations, suspicious sessions, anomalous download, and malicious content. **NOTE:** Collecting Box Shield Alerts requires implementing Box Shield,	Appends data	60 seconds	`box_shield_alerts_raw`	—
Directory and metadata					
Users	Lists user data.	Overwrites data	10 minutes	`box_users_raw`	—
Groups	Lists user group data.	Overwrites data	10 minutes	`box_groups_raw`	—

PREREQUISITE:

1. Set up an Enterprise Box plan.

   IMPORTANT:

   To collect Box Shield Alerts, you must purchase Box Shield and it must be enabled on Box enterprise.

2. Create a valid Box account that is assigned to a role with sufficient permissions for the data you want to collect. For example, create an account assigned to an Admin role to enable Cortex XSIAM to collect all metadata for all files, folders, and enterprise events for the entire organization.
3. Enable two-factor authentication for the Box account. For more information, see the Box documentation.

Configure Cortex XSIAM to receive logs and data from Box.

1. Complete the prerequisites mentioned above for your Box enterprise account.
2. Create a new app in your Box account.

a. Log in to your Box account, and in the Dev Console, click Create New App.

b. Select Custom App.

c. Set these settings in the Custom App dialog:

- Select Server Authentication (Client Credentials Grant).
- Specify an App Name.
- Click Create App.

The new app is created and the opened in the Configuration tab.

d. In the Configuration tab of the new app, scroll down to the following sections and configure the app.

- In the App Access Level section, select App + Enterprise Access.
- In the Application Scopes section, set the following Administrative Action permissions depending on the type of data you want to collect.

Administrative Action	**Data Type**
Manage users	Users
Manage groups	Groups **NOTE:** There is a current bug with the Groups API from Box. If you don't configure the Box app with the proper permissions for managing groups data, the Groups API from Box won't return an error message to Cortex XSIAM indicating that the API failed to receive the data, and the Groups data will not be collected.
Manage enterprise properties	◦ Events (admin_logs) ◦ Box Shield Alerts

Once completed, scroll up in the tab to Save Changes.

e. In the Authorization tab, click Review and Submit to send your changes to the administrator for approval.

In the Review App Authorization Submission dialog that is displayed, you can add a Description of the app changes, and then click Submit.

3. Ensure the new app changes are approved by an administrator in the Admin Console of the Box account.

a. Select Apps → Customer Apps Manager → Server Authentication Apps.

b. In the table, look for the Name of the Box app with the changes, where the Authorization Status is set to Pending Authorization, and select the options menu → Authorize App.

c. Click Authorize.

NOTE:

For any future change that you make to your Box app, ensure that you send the changes for approval to the administrator, who will need to approve them as explained above.

4. In Cortex XSIAM, select Settings → Data Sources.

5. On the Data Sources page, click Add Data Source, search for and select Box, and click Connect.

6. Set the following parameters, where some values require you to log in to your Box account to copy and paste the values to the applicable fields:

- Name: Specify a descriptive name for this Box instance.
- Enterprise ID: Specify the unique identifier for your organization's Box instance, which is used to access the token request. This field can't be edited once the Box data collector instance is created.

  You can retrieve this value from your Box account in the the General Settings tab, and scrolling to the App Info section. Copy the Enterprise ID and paste it in this field in Cortex XSIAM.
- Client ID: Specify the client ID or API key for the Box app you created.

  You can retrieve this value from your Box account in the Configuration tab, and scrolling down to the OAuth 2.0 Credentials section. COPY the Client ID and paste it into this field in Cortex XSIAM.
- Client Secret: The client secret or API secret fort he Box app you created.

  You can retrieve this value from your Box account in the Configuration tab, and scrolling down to the OAuth 2.0 Credentials section. Click Fetch Client Secret, where you will need to authenticate yourself according to the two-factor authentication method defined in your Box app before the Client Secret is displayed. Copy this value and paste it in this field in Cortex XSIAM.
- Collect: Select the types of data you want to collect from Box. All the options are selected by default.
  - Events and security alerts
    - Events (admin_logs): Collects events related to file/folder management, permission changes, access and login activities, user/groups management, folder collaboration, file/folder sharing, security settings changes, tasks, permission changes on folders, storage expiration and data retention, and workflows.
    - Box Shield Alerts: Collects security alerts related to suspicious locations, suspicious sessions, anomalous download, and malicious content.
  - Directory and metadata

    > **NOTE:**
    >
    > Inventory data snapshots are collected every 10 minutes.

    - Users: Collects user data.
    - Groups: Collects user group data.

7. To test the connection settings, click Test.
8. If the test is successful, click Enable to enable Box log collection.

   When events start to come in, a green check mark appears underneath the Box configuration.

1.6.9.11 | Ingest logs and data from Dropbox

Abstract

Ingest logs and data from Dropbox Business accounts via the Dropbox Business API.

Cortex XSIAM can ingest different types of data from Dropbox Business accounts using the Dropbox data collector. To receive logs and data from Dropbox Business accounts via the Dropbox Business API, you must configure the Data Sources settings in Cortex XSIAM based on your Dropbox Business Account credentials. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset for the different types of data that you are collecting, which you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library. For all logs, Cortex XSIAM can generate Cortex XSIAM issues (Analytics, Correlation Rules, IOC, and BIOC), when relevant, from Dropbox Business logs. While Correlation Rules issues are generated on non-normalized and normalized logs, Analytics, IOC, and BIOC issues are only generated on normalized logs.

The following table provides a brief description of the different types of data you can collect, the collection method and fetch interval for new data collected, the name of the dataset to use in Cortex XSIAM to query the data using XQL Search, and whether the data is normalized.

> **NOTE:**
>
> The Fetch Interval is non-configurable.

Type Of Data	Description	Collection Method	Fetch Interval	Dataset Name	Normalized Data
Log collection					

Type Of Data	Description	Collection Method	Fetch Interval	Dataset Name	Normalized Data
Events	Retrieves team events, including access events, administrative events, file/folders events, security settings events, and more. team_log/get_events	Appends data	60 seconds	dropbox_events_raw	When relevant, Cortex XSIAM normalizes SaaS audit event logs into stories, which are collected in a dataset called saas_audit_logs.
Directory and metadata					
Member Devices	Lists all device sessions of a team. team/devices/list_members_devices	Overwrites data	10 minutes	dropbox_members_devices_raw	—
Users	Lists members of a group. team/members/list_v2	Overwrites data	10 minutes	dropbox_users_raw	—
Groups	Lists groups on a team. team/groups/list	Overwrites data	10 minutes	dropbox_groups_raw	—

PREREQUISITE:

1. Set up an Advanced Dropbox plan.
2. Create a Dropbox Business admin account with Security admin permissions, which is required to authorize Cortex XSIAM to access the Dropbox Business account and generate the OAuth 2.0 access token.

Configure Cortex XSIAM to receive logs and data from Dropbox.

1. Complete the prerequisite steps mentioned above for your Dropbox Business account.
2. Log in to Dropbox using an admin account designated with Security admin level permissions.
3. In the Dropbox App console, ensure that you either create a new app, or your existing app is created, with the following settings:
   - Choose an API: Select Scoped access.
   - Choose the type of access you need: Select Full dropbox for access to all files and folders in a user's Dropbox.
4. In the Permissions tab of your app, ensure that the applicable permissions are selected under the relevant section heading for the type of data you want to collect:

Section Heading	Permission	Data To Collect
Account Info	account_info.read	All types of data
Team Data	team_data.member	All types of data
Members	members.read	Users
	groups.read	Groups

Section Heading	Permission	Data To Collect
Sessions	sessions.list	Member Devices
	events.read	Events

5. In the Settings tab of your app, copy the App key and App secret , where you must click Show to see the App secret and record them somewhere safe. You will need to provide these keys when you configure the Dropbox data collector in Cortex XSIAM.
6. In Cortex XSIAM, select Settings → Data Sources.
7. On the Data Sources page, click Add Data Source, search for and select Dropbox and click Connect.
8. Set the following parameters:
   - Name: Specify a descriptive name for this Dropbox instance.
   - App Key: Specify the App key, which is taken from the Settings tab of your Dropbox app.
   - App Secret: Specify the App secret, which is taken from the Settings tab of your Dropbox app.
   - Access Code: After specifying an App Key, you can obtain the access code by hovering over the Access Code tooltip, clicking the here link, and signing in with your Dropbox Business account credentials. The URL link is `https://www.dropbox.com/oauth2/authorize?client_id=%APP_KEY%&token_access_type=offline&response_type=code`, where the `%APP_KEY%` is replaced with the App Key value specified.

     > **NOTE:**
     >
     > When the App Key field is empty, the here link in the tooltip is disabled. When an incorrect App Key is entered, clicking the link results in a 404 error.

     To obtain the Access Code complete the following steps in the page that opens in your browser:

     1. Read the disclaimer and click Continue.
     2. Review the permissions listed, which should match the permissions you configured in your Dropbox app in the Permissions tab according to the type of data you want to collect, and click Allow.
     3. Copy the Access Code Generated and paste it in the Access Code field in Cortex XSIAM. The access code is valid for around four minutes from when it is generated.

     > **NOTE:**
     >
     > Whenever you change the permissions of the Dropbox app, we recommend that you generate a new Access Code for the Dropbox data collector instance so that the permissions match the updates.

   - Collect: Select the types of data you want to collect from Dropbox. All the options are selected by default.
     - Log collection
       - Events (get_events}: Retrieves team events, including access events, administrative events, file/folders events, security settings events and more.

       > **NOTE:**
       >
       > Event data is collected every 60 seconds with a 10 minute lag time.

     - Directory and metadata
       - Member Devices: Collects all device sessions of a team.
       - Users: Collects all members of a group.
       - Groups: Collects all groups on a team.

       > **NOTE:**
       >
       > Inventory data snapshots are collected every 10 minutes.
9. To test the connection settings, click Test.
10. If the test is successful, click Enable to enable Dropbox log collection.

    After events start to come in, a green check mark appears underneath the Dropbox configuration.

1.6.9.12 | Ingest logs from Elasticsearch Filebeat

Abstract

Cortex XSIAM can ingest logs from Elasticsearch Filebeat, a file system logger that logs file activity on your endpoints and servers.

If you want to ingest logs about file activity on your endpoints and servers and do not use the Cortex XDR agent, you can install Elasticsearch Filebeat as a system logger and then forward those logs to Cortex XSIAM. To facilitate log ingestion, Cortex XSIAM supports the same protocols that Filebeat and Elasticsearch use to communicate. Cortex XSIAM supports using Filebeat up to version 8.2 with the Filebeat data collector. Cortex XSIAM also supports logs in single line format or multiline format. For more information on handling messages that span multiple lines of text in Elasticsearch Filebeat, see Manage Multiline Messages.

Cortex XSIAM supports all sections in the `filebeat.yml` configuration file, such as support for Filebeat fields and tags. As a result, this enables you to use the add_fields processor to identify the product/vendor for the data collected by Filebeat so the collected events go through the ingestion flow (Parsing Rules). To configure the product/vendor ensure that you use the default `fields` attribute, as opposed to the `target` attribute, as shown in the following example.

```
processors:
  - add_fields:
      fields:
        vendor: <Vendor>
        product: <Product>
```

To provide additional context during investigations, Cortex XSIAM automatically creates a new Cortex Query Language (XQL) dataset from your Filebeat logs. You can then use the XQL dataset to search across the logs Cortex XSIAM received from Filebeat.

To receive logs, you configure collection settings for Filebeat in Cortex XSIAM and output settings in your Filebeat installations. As soon as Cortex XSIAM begins receiving logs, the data is visible in XQL Search queries.

1. In Cortex XSIAM, set up Data Collection.

    a. Select Settings → Data Sources.

    b. On the Data Sources page, click Add Data Source, search for and select Filebeat, and click Connect.

    c. Specify a descriptive Name for your Filebeat log collection configuration.

    d. Specify the Vendor and Product for the type of logs you are ingesting.

    The vendor and product are used to define the name of your XQL dataset (`<vendor>_<product>_raw`). If you do not define a vendor or product, Cortex XSIAM examines the log header to identify the type and uses that to define the vendor and product in the dataset. For example, if the type is Acme and you opt to let Cortex XSIAM determine the values, the dataset name would be `acme_acme_raw`.

    e. Save & Generate Token.

    Click the copy icon next to the key and record it somewhere safe. You will need to provide this key when you set up output settings on your Filebeat instance. If you forget to record the key and close the window you will need to generate a new key and repeat this process.

2. Set up Filebeat to forward logs.

    After installing the Filebeat agent, configure an Elasticsearch output:

    a. Under the output.elasticsearch section, configure the following entities:

output.elasticsearch:

```
output.elasticsearch:
    # Array of hosts to connect to.
    hosts: ["{URL copied from Cortex XDR}"]

    compression_level: 5

    bulk_max_size: 1000

    # Authentication credentials - either API key or use±rname/passwordo.
    api_key: "{Token generated by the Cortex XDR integration}"
```

- `hosts`: Copy the API URL from your Filebeat configuration and paste it in this field.
- `compression_level`: 5 (recommended)
- `bulk_max_size`: 1000 (recommended)
- `api_key`: Paste the key you created in when you configured Filebeat Log Collection in Cortex XSIAM.
- `proxy_url`: (Optional) `<server_ip>:<port_number>`. You can specify your own `<server_ip>` or use the Broker VM to proxy Filebeat communication using the format `<Broker_VM_ip>:<port_number>`. When using the Broker VM, ensure that you activate the Local Agent Settings applet with the Agent Proxy enabled.

b. Save the changes to your output file.

After Cortex XSIAM begins receiving logs from Filebeat, they will be available in XQL Search queries.

3. (Optional) Monitor your Filebeat integration.

   You can return to the Settings → Configurations → Data Collection → Data Sources page to monitor the status of your Filebeat configuration. For each instance, Cortex XSIAM displays the number of logs received in the last hour, day, and week. You can also use the Data Ingestion Dashboard to view general statistics about your data ingestion configurations.

4. (Optional) Set up issue notifications to monitor the following events.
   - A Filebeat agent status changes to disconnected.
   - A Filebeat module has stopped sending logs.

1.6.9.13 | Ingest logs from Forcepoint DLP

Abstract

Extend Cortex XSIAM visibility into logs from Forcepoint DLP.

If you use Forcepoint DLP to prevent data loss over endpoint channels, you can take advantage of Cortex XSIAM investigation and detection capabilities by forwarding your logs to Cortex XSIAM. This enables Cortex XSIAM to help you expand visibility into data violation by users and hosts in the organization, correlate and detect DLP incidents, and query Forcepoint DLP logs using XQL Search.

When Cortex XSIAM starts to receive logs, Cortex XSIAM can analyze your logs in XQL Search and you can create new Correlation Rules.

To integrate your logs, you first need to set up an applet in a Broker VM within your network to act as a Syslog Collector. You then configure forwarding on your log devices to send logs to the Syslog Collector in a CEF or LEEF format.

Configure Forcepoint DLP collection in Cortex XSIAM.

1. Verify that your Forcepoint DLP meet the following requirements.
   - Must use version 8.8.0.347 or a later release.
   - On premise installation only.
2. Activate the Syslog Collector applet on a Broker VM in your network.

   Ensure the Broker VM is configured with the following settings.
   - Format: Select either a CEF or LEF Syslog format.
   - Vendor: Specify the Vendor as `forcepoint`.
   - Product: Specify the Product as `dlp_endpoint`.
3. Increase log storage for Forcepoint DLP logs.

   As an estimate for initial sizing, note the average Forcepoint DLP log size. For proper sizing calculations, test the log sizes and log rates produced by your Forcepoint DLP. For more information, see Manage Your Log Storage.
4. Configure the log device that receives Forcepoint DLP logs to forward syslog events to the Syslog Collector in a CEF or LEEF format.

   For more information, see the Forcepoint DLP documentation.
5. After Cortex XSIAM begins receiving data from Forcepoint DLP, you can use XQL Search to search your logs using the `forcepoint_dlp_endpoint` dataset.

1.6.9.14 | Ingest logs from Proofpoint Targeted Attack Protection

Abstract

Ingest logs from Proofpoint Targeted Attack Protection (TAP).

To receive logs from Proofpoint Targeted Attack Protection (TAP), you must first configure TAP service credentials in the TAP dashboard, and then the Collection Integrations settings in Cortex XSIAM based on your Proofpoint TAP configuration. After you set up data collection, Cortex XSIAM begins receiving new logs and data from the source.

When Cortex XSIAM begins receiving logs, the app creates a new dataset (`proofpoint_tap_raw`) that you can use to initiate XQL Search queries. For example queries, refer to the in-app XQL Library.

Configure the Proofpoint TAP collection in Cortex XSIAM.

1. Generate TAP Service Credentials in Proofpoint TAP.

   TAP service credentials can be generated in the TAP Dashboard, where you will receive a Proofpoint Service Principal for authentication and Proofpoint API Secret for authentication. Record these credentials as you will need to provide them when configuring the Proofpoint Targeted Attack Protection data collector in Cortex XSIAM. For more information on generating TAP service credentials, see Generate TAP Service Credentials.

2. Configure the Proofpoint TAP collection in Cortex XSIAM.

   a. Select Settings → Data Sources.

   b. On the Data Sources page, click Add Data Source, search for and select Proofpoint Targeted Attack Protection, and click Connect.

   c. Set these parameters:

      - Name: Specify a descriptive name for your log collection configuration.
      - Proofpoint Endpoint: All Proofpoint endpoints are available on the `tap-api-v2.proofpoint.com` host. You can leave the default configuration or specify another host.
      - Service Principal: Specify the Proofpoint Service Principal for authentication. TAP service credentials can be generated in the TAP Dashboard.
      - API Secret: Specify the Proofpoint API Secret for authentication. TAP service credentials can be generated in the TAP Dashboard.

   d. Click Test to validate access, and then click Enable.

      Once events start to come in, a green check mark appears underneath the Proofpoint Targeted Attack Protection configuration with the amount of data received.

3. (Optional) Manage your Proofpoint Targeted Attack Protection data collector.

   After you enable the Proofpoint Targeted Attack Protection data collector, you can make additional changes as needed.

   You can perform any of the following:

   - Edit the Proofpoint Targeted Attack Protection data collector settings.
   - Disable the Proofpoint Targeted Attack Protection data collector.
   - Delete the Proofpoint Targeted Attack Protection data collector.

1.6.9.15 | Ingest logs and data from Salesforce.com

Abstract

Use the Cortex XSIAM data collector to collect Audit Trail and Security Monitoring event logs from Salesforce.com.

The Cortex XSIAM data collector can collect Audit Trail and Security Monitoring event logs from Salesforce.com. During setup of this data collector, you can choose to accept the default collection settings, or exclude the collection of content metadata and accounts.

The Salesforce.com data collector fetches events, and objects and metadata, including:

- Login history
- Setup audit trail
- Flow Execution events
- Transaction Security events
- Content Distribution events
- Package Install events

You can create multiple Salesforce.com data collector instances in Cortex XSIAM, for different parts of your organization.

Logs are collected from Salesforce.com every 30 seconds. When Cortex XSIAM begins receiving logs, it creates new datasets for them, called `salesforce_<object>_raw`. Examples of `<object>` include:

- connectedapplication
- permissionset
- profile
- groupmember
- group
- user
- userrole
- document
- contentfolder
- attachment
- contentdistribution
- tenantsecuritylogin
- useraccountteammember
- tenantsecurityuserperm
- account
- audit
- login
- eventlogfile

You can use these datasets to perform XQL search queries. For example queries, refer to the in-app XQL Library.

> **PREREQUISITE:**

- Cortex XSIAM:
  - To manage collection integration in Cortex XSIAM, ensure that you have the privilege to View/Edit Log Collections (for example, Instance Administrator).
- Salesforce.com:
  - The minimum required Salesforce.com editions are Professional Edition with API access enabled, or Enterprise Edition, or higher.
  - To use the client credentials flow required for Salesforce.com–Cortex XSIAM integration, you must create a connected app for Cortex XSIAM in Salesforce.com, and configure its OAuth settings and access policies, as described in this procedure. The connected app must be created by a Full System Admin.
  - Ensure that your organization has a Salesforce Shield license. For more information, refer to:

    https://help.salesforce.com/s/articleView?id=xcloud.salesforce_shield.htm&type=5

    https://trailhead.salesforce.com/content/learn/modules/event_monitoring/event_monitoring_intro

    NOTE:

    Ensure that you have the required licenses. If these prerequisites are not met, fetching of security data and event data will be severely limited, and errors will be generated.
  - In Setup, Event Monitoring Settings, ensure that Generate event log files is enabled.
    - In Setup, verify that there are event log files in the Event Log File Browser.
    - In Setup, Permission Sets, verify that there is a permission set called Event Monitoring.

NOTE:

For more detailed reference information, see Configure a Connected App for the OAuth 2.0 Client Credentials Flow.

Unlike other data collector setups, in this case, the setup includes obtaining an OAuth 2.0 code from Salesforce.com, and this code is only valid for 15 minutes. Therefore, make sure that you enable the data collector within 15 minutes of obtaining the authorization code.

Perform the following procedures in the order that they appear, below.

Task 1. Configure Salesforce Connected App

1. On the Setup page, in Quick Find, type `App Manager`.
2. Click New Connected App.
3. Enter a meaningful name for the connected application and for the API. For example, you could name it panw_cortex_integration.
4. Enter your email address. This address will be used to retrieve the Consumer Key and Consumer Secret.
5. Select the Enable OAuth Settings checkbox.
6. In Callback URL, type

   `https://login.salesforce.com/services/oauth2/callback`

   and

   `https://{tenant external URL}.paloaltonetworks.com/configuration/data-sources`

   on separate lines, where `{tenant external URL}` is the name of your tenant as it appears in the URL of your Cortex XSIAM tenant.
7. For OAuth Scopes, select Full access (full) and Perform requests at any time (refresh_token, offline_access).
8. In the next options after OAuth Scopes, ensure that only the following checkboxes are selected:
   - Require Secret for Web Server Flow
   - Require Secret for Refresh Token Flow
   - Enable Credentials Flow
9. Click Save, and then Continue.

Task 2. Retrieve the Consumer Key and Consumer Secret

Consumer Key will be used for client_id, and Consumer Secret will be used for client_secret in OAuth 2.0.

1. On the Setup page, in Quick Find, type `App Manager`.
2. Find your connected application (the one that you defined for Cortex XSIAM). In the last column, click the arrow button and then click View.
3. In the API (Enable OAuth Settings) area, click Manage Consumer Details.

4. When you are asked to verify your identity, open the email that Salesforce sent to you, and copy the verification code. Go back to the Salesforce Verify Your Identity page, paste the code in the Verification Code box, and click Verify. One of the following will happen:
    - The Consumer Key and Consumer Secret will be sent to the email address that you configured earlier for the Cortex XSIAM connected app.
    - On the Salesforce Connected App Name page, the Consumer Details area will display the Consumer Key and Consumer Secret, and you will be able to copy them from here when required in the following procedures.

Task 3. Configure the Refresh Token expiration policy

1. On the Setup page, in Quick Find, type `App Manager`.
2. Find your connected application (the one that you defined for Cortex XSIAM). In the last column, click the arrow button and then click Manage.
3. Click Edit Policies.
4. In the OAuth Policies area:
    - Under Permitted Users, select All users may self-authorize.
    - Choose your refresh token policy. We recommend: Expire refresh token if not used for _ Day(s). For example, select this option and set it for 7 days.

Task 4. Configure OAuth 2.0

- Configure the OAuth 2.0 application to call the Salesforce.com API using client_id and client_secret.

  References: https://help.salesforce.com/s/articleView?id=sf.remoteaccess_oauth_client_credentials_flow

Task 5. Configure Cortex XSIAM

1. In Cortex XSIAM, create a Salesforce.com data collector instance:
    - Select Settings → Data Sources.
    - On the Data Sources page, click Add Data Source, search for and select Salesforce.com, and click Connect.
2. Enter a unique Name for the instance, enter the Salesforce Domain Name, and the Consumer Key and the Consumer Secret credentials obtained earlier in this workflow. For example, the domain could be the API URL from which logs are received, such as `https://MyDomainName.my.salesforce.com/services/data/vXX.X/resource/`
3. (Optional) Clear options that you do not require:
    - Content metadata: when selected (default), collects documents' metadata.
    - Accounts: when selected (default), collects account objects.

    > **NOTE:**
    >
    > When these options are cleared, only these data types will be omitted from collection. All other data will be collected as usual.
4. Click Enable. A popup which redirects you to your Salesforce instance appears, to get OAuth 2.0 authorization credentials and access.
5. Click OK.

   In Salesforce.com, a new tab appears.
6. Enter your username and password, and Log In.
7. When you are asked to allow access, select Allow.

   A Salesforce data collection instance is created, and an authorization token is created and returned to Cortex XSIAM. Data collection begins.

Task 6. (Optional) Edit or test existing Salesforce.com collector settings

You can edit and test an existing collector instance after a successful initial connection between Salesforce.com and Cortex XSIAM. Do this by clicking Edit (pencil icon) for the collector instance. The log collection window will be displayed, where you can make changes or test, by clicking Test.

Troubleshooting

If for any reason, the token is not created and sent to Cortex XSIAM, after a timeout period, an authorization failure error will be returned for the collector instance. In this case, try again by clicking Edit (pencil icon) for the collector instance. The log collection window will be displayed again, where you can edit settings and retry getting the authorization code.

1.6.9.16 | Ingest data from ServiceNow CMDB

Abstract

Extend Cortex XSIAM visibility into data from ServiceNow CMDB.

To receive data from the ServiceNow CMDB database, you must first configure data collection from ServiceNow CMDB. ServiceNow CMDB is a logical representations of assets, services, and the relationships between them that comprise the infrastructure of an organization. It is built as a series of connected tables that contain all the assets and business services controlled by a company and its configurations. You can configure the Collection Integration settings in Cortex XSIAM for the ServiceNow CMDB database, which includes selecting the specific tables containing the data that you want to collect, in the ServiceNow CMDB Collector. You can select from the list of default tables and also specify custom tables. By default, the ServiceNow CMDB Collector is configured to collect data from the following tables, which you can always change depending on your system requirements.

- `cmdb_ci`
- `cmdb_ci_computer`
- `cmdb_rel_ci`
- `cmdb_ci_application_software`

When Cortex XSIAM begins receiving data, the app automatically creates a ServiceNow CMDB dataset for each table using the format `servicenow_cmdb_<table name>_raw`. You can then use XQL Search queries to view the data and create new Correlation Rules.

You can only configure a single ServiceNow CMDB Collector, which is automatically configured every 6 hours, to reload the data from the configured tables and replace the existing data. You can always use the Sync Now option to reload the data and replace the existing data whenever you want.

Complete the following task before you begin configuring Cortex XSIAM to receive data from ServiceNow CMDB.

- Create a ServiceNow CMDB user with SNOW credentials, who is designated to access the tables from ServiceNow CMDB for data collection in Cortex XSIAM. Record the credentials for this user as you will need them when configuring the ServiceNow CMDB Collector in Cortex XSIAM.

Configure Cortex XSIAM to receive data from ServiceNow CMDB:

1. Select Settings → Data Sources.
2. On the Data Sources page, click Add Data Source, search for and select ServiceNow CMDB, and click Connect.
3. Set the following parameters.
    - Domain: Specify your ServiceNow CMDB domain URL.
    - User Name: Specify the username for your ServiceNow CMDB user designated in Cortex XSIAM.
    - Password: Specify the password for your ServiceNow CMDB user designated in Cortex XSIAM.
    - Tables: You can do any of the following actions to configure the tables whose data is collected from ServiceNow CMDB.
        - Select the tables from the list of default ServiceNow CMDB tables that you want to collect from. After each table selection, select ↩ to add the table to the tables already listed below for data collection.
        - Specify any custom tables that you want to configure for data collection.
        - From the default list of tables already configured, you can delete any of them by hovering over the table and selecting the X icon.
4. Click Test to validate access, and then click Enable.

    After events start to come in, a green check mark appears underneath the ServiceNow CMDB Collector configuration with the data and time that the data was last synced.

5. (Optional) Manage your ServiceNow CMDB Collector.

    After you enable the ServiceNow CMDB Collector, you can make additional changes as needed. To modify a configuration, select any of the following options:

    - Edit the ServiceNow CMDB Collector settings.
    - Disable the ServiceNow CMDB Collector.
    - Delete the ServiceNow CMDB Collector.
    - Sync Now to get the latest data from the tables configured. The data is replaced automatically every 6 hours, but you can always get the latest data as needed.
6. After Cortex XSIAM begins receiving data from ServiceNow CMDB, you can use the XQL Search to search for logs in the new datasets, where each dataset name is based on the table name using the format `servicenow_cmdb_<table name>_raw`.

1.6.9.17 | Ingest report data from Workday

Abstract

Extend Cortex XSIAM visibility into reports data from Workday.

To receive Workday report data, you must first configure data collection from Workday using a Workday custom report to ingest the appropriate data. This is configured by setting up a Workday Collector in Cortex XSIAM and configuring report data collection via this Workday custom report that you set up.

As soon as Cortex XSIAM begins receiving data, the app automatically creates a Workday Cortex Query Language (XQL) dataset (`workday_workday_raw`). You can then use XQL Search queries to view the data and create new Correlation Rules. In addition, Cortex XSIAM adds the Workday fields next to each user in the Key Assets list on the Cases page, and in the User node in the Causality View of Identity Analytics issues.

NOTE:

Any user with permissions to view issues and cases can view the Workday data.

You can only configure a single Workday Collector, which is automatically configured to run the report every 6 hours. You can always use the Sync Now option to run the report whenever you want.

PREREQUISITE:

1. Create an Integration System User that is designated to access the custom report from Workday for data collection in Cortex XSIAM.
2. Create an Integration System Security Group for the Integration System User created in Step 1 for accessing the report. When setting this group ensure to define the following:
   - Type of Tenanted Security Group: Select either Integration System Security Group (Constrained) or Integration System Security Group (Unconstrained) depending on how your data is configured. For more information, see the Workday documentation.
   - Integration System User: Select the user that you defined in step 1 for accessing the custom report.
3. Create the Workday credentials for the Integration System User created in Step 1 so that the username and password can be used to access the report in Cortex XSIAM. Record these credentials as you will need them when configuring the Workday Collector in Cortex XSIAM.

NOTE:

For more information on completing any of the prerequisite steps, see the Workday documentation.

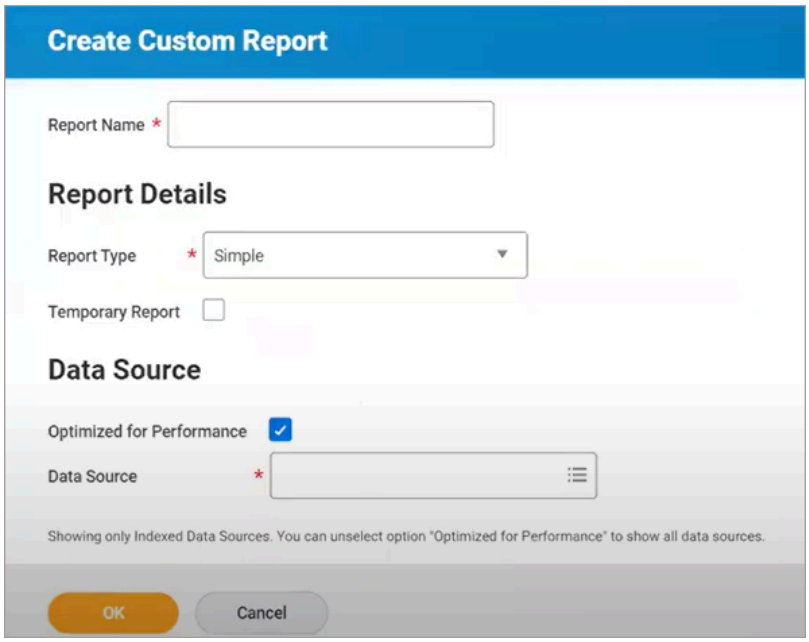
Configure Cortex XSIAM to receive report data from Workday:

1. Configure a Workday custom report to use for data collection.

   a. Login to the Workday Resource Center.

   b. In the search field, specify Create Custom Report to open the wizard.

   c. Configure the following Create Custom Report settings:

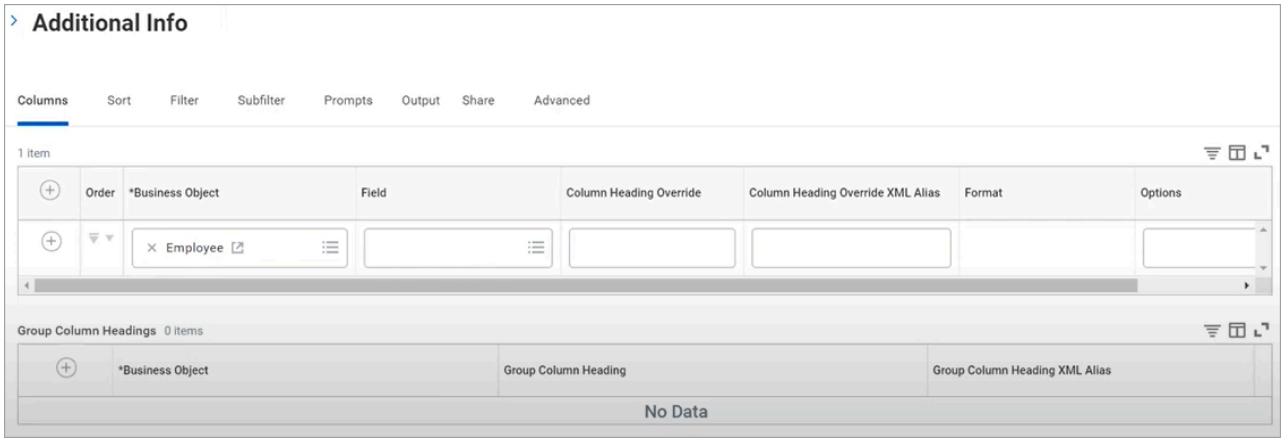


- Report Name: Specify the name of the report.
- Report Details section:
  - Report Type: Select Advanced. When you select this option, the Enable As Web Service checkbox is displayed.
  - Enable As Web Service: Select this checkbox, so that you will be able to generate a URL of the report to configure in Cortex XSIAM.
- Data Source section:
  - Optimized for Performance: Select whether the data should be optimized for performance. The way this checkbox is configured determines the Data Source options available to choose from.
  - Date Source: Select the applicable data source containing the data that is used to configure data collection from Workday to Cortex XSIAM.

d. Click OK, and configure the following Additional Info settings.

The Additional Info table in the Columns tab is where you can perform the following.

- For the incident and card views in Cortex XSIAM, map the required fields from the Data Source configured by selecting the applicable Field that you want to map to the Cortex XSIAM field name required for data collection in the Column Heading Override XML Alias column.
- (Optional) You can map any additional fields from the Data Source configured that you want to be able to query in XQL Search using the `workday_workday_raw` dataset. This is configured by selecting the applicable Field and leaving the default field name that is displayed in the Column Heading Override XML Alias column. This default field name is what is used in XQL Search and the dataset to view and query the data.



> **NOTE:**
>
> The Business Object changes depending on the Data Source selected.

For the incident and card views in Cortex XSIAM, map the following fields in the table by selecting the applicable Field that contains the data representing the Cortex XSIAM field name as provided below that should be added to the Column Heading Override XML Alias. For example, for `full_name`, select the applicable Field from the Business Object defined that contains the full name of the user and in the Column Heading Override XML Alias specify `full_name` to map the set Field to the Cortex XSIAM field name.

> **NOTE:**
>
> Cortex XSIAM uses a structured schema when integrating Workday data. To get the best Analytics results, specify all the fields marked with an asterisk from the recommended schema.

- `workday_user_id*`
- `full_name*`
- `workday_manager_user_id*`
- `manager*`
- `worker_type*`
- `position_title*`
- `department*`
- `private_email_address*`
- `business_email_address*`
- `employment_start_date*`
- `employment_end_date`
- `phone_number`
- `mailing_address`

e. (Optional) Filter out any employees that you do not want included in the Filter tab.

f. Share access to the report with the designated Integration System User that you created by setting the following settings in the Share tab:

- Report Definition Sharing Options: Select Share with specific authorized groups and users.
- Authorized Users: Select the designated Integration System User that you created for accessing the custom report.

g. Ensure that the following Web Services Options settings in the Advanced tab are configured.

Here is an example of the configured settings, where the Web Service API Version and Namespace are automatically populated and dependent on your report.

h. (Optional) Test the report to ensure all the fields are populated.

i. Get the URL for the report.

1. In the related actions menu, select Actions → Web Service → View URLs.
2. Click OK.
3. Scroll down to the JSON section.
4. Hover over the JSON link and click the icon, which open a new tab in your browser with the URL for the report. You need to use the designated user credentials to open the report.
5. Copy the URL for the report and record them somewhere as this URL needs to be provided when setting up the Workday Collector in Cortex XSIAM.

j. Complete the report by clicking Done.

2. Configure the Workday collection in Cortex XSIAM.

a. Select Settings → Data Sources.

b. On the Data Sources page, click Add Data Source, search for and select Workday, and click Connect.

c. Set the following parameters.

- Name: Specify the name for the Workday Collector that is displayed in Cortex XSIAM.
- URL: Specify the URL of the custom report you configured in Workday.
- User Name: Specify the username for the designated Integration System User that you created for accessing the custom report in Workday.
- Password: Specify the password for the designated Integration System User that you created for accessing the custom report in Workday.

d. Click Test to validate access, and then click Enable.

A notification appears confirming that the Workday Collector was saved successfully, and closes on its own after a few seconds.

Once report data starts to come in, a green check mark appears underneath the Workday Collector configuration with the data and time that the data was last synced.

3. (Optional) Manage your Workday Collector.

After you enable the Workday Collector, you can make additional changes as needed. To modify a configuration, select any of the following options.

- Edit the Workday Collector settings.
- Disable the Workday Collector.
- Delete the Workday Collector.
- Sync Now to run the report to get the latest report data. The report is run automatically every 6 hours, but you can always get the latest data as needed.

4. After Cortex XSIAM begins receiving report data from Workday, you can use the XQL Search to search for logs in the new dataset (`workday_workday_raw`).

1.6.9.18 | Ingest external alerts

Abstract

For a more complete and detailed picture of the activity involved in a case, Cortex XSIAM can ingest alerts from any external source.

For a more complete and detailed picture of the activity involved in a case, Cortex XSIAM can ingest alerts from any external source. Cortex XSIAM stitches the external alerts together with relevant endpoint data and displays alerts from external sources in relevant cases and issues tables. You can also see external alerts and related artifacts and assets in causality views. For example, in the Issues table, right-click an issue and select Investigate Causality Chain.

To ingest alerts from an external source, you configure your alert source to forward alerts (in Auto-Detect (default), CEF, LEEF, CISCO, or CORELIGHT format) to the Syslog collector. You can also ingest alerts from external sources using the Cortex XSIAM APIs.

After Cortex XSIAM begins receiving external alerts, you must map the following required fields to the Cortex XSIAM format.

- TIMESTAMP
- SEVERITY
- ALERT NAME

In addition, these optional fields are available, if you want to map them to the Cortex XSIAM format.

- SOURCE IP
- SOURCE PORT
- DESTINATION IP
- DESTINATION PORT
- DESCRIPTION
- DIRECTION
- EXTERNAL ID
- CATEGORY
- ACTION
- PROCESS COMMAND LINE
- PROCESS SHA256
- DOMAIN
- PROCESS FILE PATH
- HOSTNAME
- USERNAME

NOTE:

If you send pre-parsed alerts using the Cortex XSIAM API, additional mapping is not required.

Storage of external alerts is determined by your Cortex XSIAM tenant retention policy. For more information, see Dataset Management.

1. Send alerts from an external source to Cortex XSIAM.

   There are two ways to send alerts:

   - API: Use the Insert CEF Alerts API to send the raw Syslog alerts or use the Insert Parsed Alerts API to convert the Syslog alerts to the Cortex XSIAM format before sending them to Cortex XSIAM. If you use the API to send logs, you do not need to perform the additional mapping step in Cortex XSIAM.
   - Activate the Syslog collector (see Activate the Syslog collector) and then configure the alert source to forward alerts to the Syslog collector. Then configure an alert/issue mapping rule as follows.

2. In Cortex XSIAM, select Settings → Configurations → Data Collection → External Issue Mapping.
3. Right-click the Vendor Product for your issues and select Filter and Map.
4. Use the filters at the top of the table to narrow the results to only the alerts you want to map.

   Cortex XSIAM displays a limited sample of results during the mapping rule creation. As you define your filters, Cortex XSIAM applies the filter to the limited sample but does not apply the filters across all alerts. As a result, you might not see any results from the alert sample during the rule creation.

5. Click Next to begin a new mapping rule.

   On the left, configure the following:

   a. Rule Information: Define the NAME and optional DESCRIPTION to identify your mapping rule.

   b. Issues Field: Map each required and any optional Cortex XSIAM field to a field in your alert source.

      If needed, use the field converter () to translate the source field to the Cortex XSIAM syntax.

      For example, if you use a different severity system, you need to use the converter to map your severities fields to the Cortex XSIAM risks of Critical, High, Medium, and Low.

      You can also use regex to convert the fields to extract the data to facilitate matching with the Cortex XSIAM format. For example, if you need to map the port, but your source field contains both the IP address and port (`192.168.1.200:8080`), to extract everything after the `:`, use the following regex:

      ```
      ^[^:]*_
      ```

      For additional context when you are investigating a case, you can also map additional optional fields to fields in your alert source.

6. To submit your alert filter and mapping rule when finished, click Submit.

1.7 | Onboard the Kubernetes Connector

Abstract

To onboard your Kubernetes cluster, specify the connection method and settings and download the custom installer file. Execute the file in your Kubernetes environment to grant Cortex XSIAM permissions to collect the data.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on.

Follow this wizard to deploy your Kubernetes Connector. The Kubernetes onboarding wizard is designed to facilitate the seamless setup of Kubernetes data into Cortex XSIAM. The guided experience requires minimal user input; simply enter a name for the installer file and select the type of connector you want to install. For full control of the setup, you can use the advanced settings. Based on the onboarding settings, Cortex XSIAM then creates a custom installer file for running in your Kubernetes environment. This file, once executed in your Kubernetes environment, grants Cortex XSIAM the necessary permissions to collect the data. The installer file must be executed in your Kubernetes environment to complete the onboarding process. The connector then appears in Kubernetes Connectors.

1. Select Settings → Data Sources.
2. Select Add Data Source.
3. On the Add Data Sources page, search for and select Kubernetes and click Connect.
4. In the Connect Kubernetes onboarding wizard, enter a name for the installer file, the deployment YAML script that is generated by the selections you choose in this wizard.
5. Select the Connector:
   - Connector: A lightweight solution that provides additional Kubernetes related capabilities, such as enhanced inventory with relations mapping and policy enforcement.
6. (Optional) Click Show advanced settings to define advanced settings:
   - Connector Namespace: Specify the Kubernetes Connector namespace.
   - Scan Cadence: Define how often to scan (from every one to 24 hours). Default is 12 hours. To optimize performance and minimize I/O impact in production environments, we recommend configuring the scan cadence to every 12 hours. This ensures comprehensive file system scanning while efficiently managing system resources.
   - Admission Controller: Select to allow enforcement policies to be configured, ensuring that only compliant resources are admitted into the cluster.
   - Version: Select which version of the Kubernetes Connector to install. For detailed information on each version, see What's new in Kubernetes Connector?.
   - Auto Upgrade: Select whether to have the Kubernetes Connector automatically upgraded when new versions become available.

   > **NOTE:**
   >
   > For GKE or EKS clusters with the metadata service disabled and for AKS clusters with non-default nodes managed resource group, the cluster resource identifier must be specified.
7. Click Next.
8. To complete the onboarding of the Kubernetes Connector, you must download the Helm chart values `values.yaml` and run it in your Kubernetes environment: `helm repo add cortex https://paloaltonetworks.github.io/cortex-cloud --force-update`
9. Install the Helm charts in your Kubernetes environment: `helm install konnector cortex/konnector --create-namespace --namespace pan --values test.values.yaml`
10. Verify the deployment succeeded when you see "Status: Deployed"

    When the Kubernetes Connector is deployed, the initial discovery scan is started and the connector appears in Data Sources → Kubernetes → Kubernetes Connectors.

1.7.1 | What's new in Kubernetes Connector?

This topic describes the changes, additions, known issues, and fixes for each version of the Kubernetes Connector.

> **LICENSE TYPE:**
>
> Requires the Cortex Cloud Posture Management add-on.

Kubernetes Connector version 1.0

New features

The following section describes the new features introduced in Kubernetes Connector version 1.0.

Feature	Description
Kubernetes Connector Onboarding	Supports AKS, EKS, and GKE clusters, with a Kubernetes Native installation method of Helm Installer.
Kubernetes Resource Discovery	Automatically discovers in-cluster resources such as namespaces, workloads, and more.
Kubernetes Policy Management & Prevention	Define and enforce policies at the Admission Controller level with prevention capabilities.
Kubernetes Compliance & Misconfiguration Detection	Leverages hundreds of out-of-the-box KSPM rules. (More pre-defined rules are being added continuously.)
Kubernetes Custom Compliance/Misconfiguration Rules Support	• Create custom compliance/misconfiguration rules using Rego (for ADS/Kubernetes Connector). • Create custom compliance/misconfiguration rules using Python (for XDR for Cloud Agent-managed Kubernetes endpoints).
KSPM Dashboard v1	A visual overview of your Kubernetes security posture. It includes inventory insights, protection coverage, riskiest clusters, and more.

Known limitations

The following table describes known limitations in the Kubernetes Connector release.

Feature	Description
Connector onboarding and cluster identifier	The Kubernetes Connector automatically calculates the Kubernetes cluster cloud identifier by using the metadata service (for EKS and GKE) and cluster resources (for AKS). • For EKS and GKE, the metadata service must be enabled. • For AKS, the cluster's node pool resource group should be the default. For more information, see Azure Docs | Create a node pool.

1.8 | Automation and feed integrations

Abstract

Set up an integration instance and start ingesting cases/indicators.

Integrations are mechanisms through which Cortex XSIAM connects and communicates with other products. These integrations can be executed through REST APIs, webhooks, and other techniques. Integrations enable you to orchestrate and automate SOC operations.

Integrations installed from a content pack

Integrations are included in content packs, which you download and install from Marketplace. After you download and install a content pack that includes an integration, you need to configure the integration by adding an instance. You can have multiple instances of an integration, for example, to connect to different environments. Additionally, if you are an MSSP and have multiple tenants, you could configure a separate instance for each tenant.

> **NOTE:**

- Some integrations can be downloaded directly without having to initially download a content pack from Marketplace. For more information, see Define data sources.
- In addition to content packs that you install from Marketplace, related content packs are automatically downloaded when you adopt playbooks or edit tasks that require content items such as scripts or integrations.

Cortex XSIAM comes out-of-the-box with integrations to help you onboard, such as:

- Mail Sender

  Sends email notifications to users.

- Generic Export Indicators Service

  Provides an endpoint with a list of indicators as a service for the system indicators. For more information about how to set up the integration, see Export indicators using the Generic Export Indicators Integration.

- Palo Alto Networks WildFire Reports

  Generates a Palo Alto Networks WildFire PDF report. For more information, see Palo Alto Networks WildFire Reports.

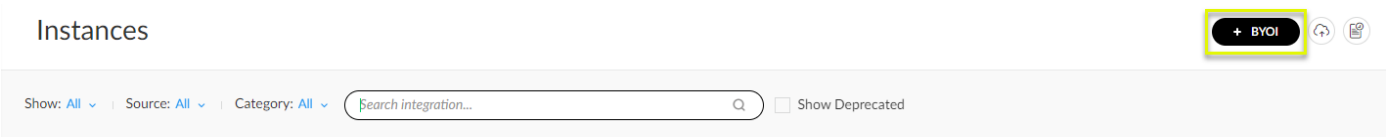
- Rasterize

  Converts URLs, PDF files, and emails to an image file or PDF file. For more information, see Rasterize.

Create an integration

You can create an integration, by adding parameters, commands, arguments, and outputs as well as writing the necessary integration code. You should have a working Cortex XSIAM tenant and programming experience with Python.

To create an integration, on the Automation and feed integrations page, click BYOI.



The Cortex XSIAM IDE and the HelloWorld integration template are loaded by default. For more information about how to create an integration, including an example, see Create an Integration.

Configure an integration

On the Automation and feed integrations page, after you have either downloaded the integration or created an integration, you can do the following:

Option	Description
Add instance	Configure an integration instance to connect and communicate with other products. For more information, see Add an integration instance. After configuring the instance, you can also enable/disable the integration instance, copy the instance, and view the integration fetch history.
View integration's source	View the integration settings and source code.
Edit integration's source	Edit the integration settings and source code. For more information about editing the integration's source code, see Create an Integration. **NOTE:** If the integration was installed from a content pack, you need to duplicate the integration before editing.
Duplicate integration	If you want to change the source code, and settings, or download the integration, you need to duplicate the integration.
Delete	Although you can't delete an integration installed from a content pack (unless a duplicate), you can delete an integration instance.

Option	Description
Always / On Demand	For each integration instance, you have the option of setting the instance to be used only On Demand, when it is specified with the `using` argument in a playbook or the CLI. By default, the settings is Always and the integration instance is used whenever the integration is called.
Download the integration	Download the integration in YAML format. You can also upload an integration. **NOTE:** If the integration was installed from a content pack, you need to duplicate the integration before downloading.
Version History	If the integration is a duplicate or you create your own integration, you can see the changes in the integration.

You can view all the integration changes (the last 100 changes) by clicking the Version History button.

Using integration commands

The command line interface (CLI) enables you to run system commands, integration commands, scripts, etc from the Cases War Room, Issues War Room, or Playground CLI. The CLI auto-complete feature allows you to find relevant commands, scripts, and arguments.

Cortex XSIAM uses the "`!`" such as `!ad-create-user username=[name of user]`

Under each integration, you can view a list of commands.

> **NOTE:**
>
> Integration commands are only available when the integration instance is enabled. Some commands depend on a successful connection between Cortex XSIAM and third-party integrations.

You can run the CLI commands in the Playground or in a case/issue War Room. The Playground is a non-production environment where you can safely develop and test automation scripts, APIs, commands, etc. It is an investigation area that is not connected to a live (active) investigation.

When running the command, the results are returned in the War Room or Playground and also in a JSON format in Context Data.

> **TIP:**
>
> In the Playground, you can clear the context data, if needed, which deletes everything in the Playground context data, but does not affect the actual issue or case. To clear the context, run `!DeleteContext all=yes'` from the CLI or click Clear Context Data while viewing the context data.

1.8.1 | Integration use cases

Abstract

Common integration use cases for Cortex XSIAM, including analytics and SIEM, authentication, case management, data enrichment, threat intelligence, forensics and malware.

The following categories are common use cases for Cortex XSIAM integrations. While this list is not meant to be exhaustive, it's a starting point to understand what use cases are supported by Cortex XSIAM and third-party integrations.

Analytics and SIEM

Top use cases:

- Fetch issues with relevant filters.
- Create, close, and delete issues/events/cases.
- Update issues - update status, assignees, severity, SLA, and more.
- Get events related to an issue/case for enrichment/investigation purposes.
- Query SIEM (consider aggregating logs).

These integrations usually include the Fetch Issues or Fetch Alerts option for an integration instance configuration. The integration may also include integration commands enabling you to list or retrieve issues or related information.

Analytics & SIEM integration Example: ArcSight ESM

Authentication and Identity Management

Top use cases:

- Use credentials from the authentication vault to configure instances in Cortex XSIAM. (Save credentials in: Settings → Configurations → Integrations → Credentials.) Integrations that use credentials from the vault should have the Switch to credentials option.
- Lock/Delete Account – Use an integration to lock/unlock a third-party account.
- Reset Account - Perform a reset password command for a third-party account.
- Lock an external credentials vault - in case of an emergency (if the vault has been compromised), allow the option to lock/unlock the entire vault via an integration.
- Step-Up authentication - Enforce Multi-Factor Authentication for an account.
- Create, update, and delete users.
- Manage user groups.
- Block users, force a change of passwords.
- Manage access to resources and applications.
- Create, update, and delete roles.

Authentication integration example: CyberArk AIM v2 (Partner Contribution)

Case Management

Top use cases:

- Create, get, edit, close a ticket or issue, and add and view comments.
- Assign a ticket/issue to a specified user.
- List all tickets, and filter by name, date, and assignee.
- Get details about a managed object, update, create, or delete.
- Add and manage users.

Case Management/Ticketing integration example: ServiceNow V2

Data Management and Threat Intelligence

Top use cases:

- Enrich information about different IOC types: Upload object for scan and get the scan results. (If there's an option to upload private/public, the default should be set to private.) Search for former scan results about an object to get information about a sample without uploading it yourself. Enrich information and scoring for the object.
- Add indicators to the system and search for existing indicators.
- Add indicators to the exclusion list.
- Calculate DBot Score for indicators.
- Enrich asset – get vulnerability information for an asset (or a group of assets) in the organization.
- Generate/trigger a scan on specified assets.
- Get a scan report including vulnerability information for a specified scan and export it.
- Get details for a specified vulnerability.
- Scan assets for a specific vulnerability.

Data Enrichment & Threat Intelligence integration example: Unit 42 Objects Feed.

Email

Top use cases:

- Get message – download the email itself, retrieve metadata, and body.
- Download attachments for a given message.
- Manage senders – block/allow specified mail senders.
- Manage URLs – block/allow the sending of specified URLs.
- Encode/decode URLs in messages
- Release a held message when a gateway has placed a suspicious message on hold.

Email Gateway integration example: MimeCast v2

Endpoint

Top use cases:

- Fetch issues and events
- Get event details (from a specified alert)
- Quarantine a file
- Isolate and contain endpoints
- Update indicators (for example, network and hashes) by policy (can be block, monitor) – deny list
- Add indicators to the exclusion list
- Search for indicators in the system (see indicators and related issues/events)
- Download a file based on the hash and the path
- Trigger scans on specified hosts
- Update .DAT files for signatures and compare existing .DAT files to the newest one on the Cortex XSIAM tenant
- Get information for a specified host (OS, users, addresses, hostname)
- Get policy information and assign policies to endpoints

Endpoint integration example: Tanium V2

Forensics and Malware Analysis

Top use cases:

- Submit a file and get a report (detonation)
- Submit a URL and get a report (detonation)
- Search for past analysis (input being a hash/URL)
- Retrieve a PCAP file
- Retrieve screenshots taken during analysis

Forensic and Malware Analysis example: Cuckoo Sandbox

Network Security

Top use cases:

- Create block/accept policies (source, destination, port), for IP addresses and domains
- Add addresses and ports (services) to predefined groups, create groups, and more
- Support custom URL categories
- Fetch network logs for a specific address for a configurable time frame
- URL filtering categorization change request
- Built-in blocked rule command for fast blocking
- If there is a Management Firewall, allow the option to manage policy rules through it
- Get/fetch issues
- Get PCAP file, packet
- Get network logs filtered by time range, IP addresses, ports, and more
- Create/manage/delete policies and rules
- Update signatures from an online source/upload + get the last signature update information
- Install policy (if existing)

Network Security Firewall integration examples: Tufin (Partner Contribution), Protectwise

Vulnerability Management

Top use cases:

- Enrich asset – get vulnerability information for an asset (or a group of assets) in the organization.
- Generate/trigger a scan on specified assets
- Get a scan report including vulnerability information for a specified scan and export it
- Get details for a specified vulnerability
- Scan assets for a specific vulnerability

Vulnerability Management integration example: Tenable.sc

1.8.2 | Add an integration instance

Abstract

Set up an integration instance and start ingesting issues/indicators.

Configure an integration instance to connect and communicate with other products.

When you define an integration instance for your third-party security and incident management vendors, events triggered by this integration instance can become cases in Cortex XSIAM. When cases are created, you can run playbooks on them to enrich them with information from other products in your system. For indicators, you can enrich those indicators depending on the integration instance and add them to a case if required.

Although you can view the integration documents when adding an instance, the Developer Hub has more detailed information about the integrations, including commands, outputs, and recommended permissions. You can also see more information about content packs, playbooks, scripts, and Marketplace documentation.

> **NOTE:**
>
> This procedure describes how to add an integration instance from the Automation and Feed Integration page. Some integration instances can also be configured on the Data Sources page. For more information, see Add a new data source or instance.

Before you begin

- From Marketplace, download and install the relevant content pack, which includes your integration. Content packs containing integrations are also downloaded, in some cases, when you adopt playbooks and configure playbook tasks.
- Consider whether you want to add credentials, which enable you to save login information without exposing usernames, passwords, certificates, and SSH keys. For more information, see Manage credentials.

1. Go to Settings → Configurations → Data Collection → Automation & Feed Integrations and search for the integration.
2. In the integration you want to add, click Add instance.
3. Add the parameters, as required.

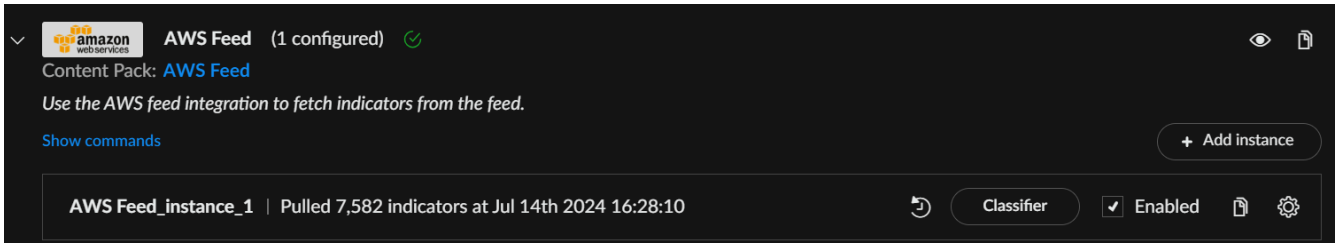
4. If you want to fetch issues, select the Fetches alerts.

   For more information, see Fetch issues from an integration instance.

5. (Optional) To check that the integration instance is working correctly, click Test.

6. Save & Exit.

   Expand the integration to see more details such as the number of pulled issues/indicators or error messages.



   You can also enable/disable the integration instance, copy the instance, and view the integration fetch history.

   If you encounter an error, see Troubleshoot Integrations.

7. By default, the integration instance is used whenever the integration is called. If you want to only use the integration instance when specified with the `using` argument in a playbook or the CLI, change the integration instance setting from Always to On Demand. For example, you might have two instances of an integration and want to use one instance as the default and the other instance only for manual testing on demand.

8. (Optional) To manage access to specific commands, see Configure integration permissions.

Example 163.

In this example, you will set up the OnboardingIntegration.

If you have not done so, download the OnboardingIntegration content pack from Marketplace. Most integrations follow a similar configuration.

1. Go to Settings → Configurations → Data Collection → Automation & Feed Integrations and search for OnboardingIntegration.

2. Click Add Instance.

3. Add the number of issues to fetch per minute. By default, there is a maximum of 5 issues per minute.

4. Add the maximum number of issues to create. By default, there is a maximum number 10 issues to create.

5. Add the number of issues you want to create in minutes.

6. Set the Alerts Fetch Interval. By default, the issues are fetched every minute.

7. Select whether to run on an engine.

8. When troubleshooting the instances, adjust the default setting from off to a higher debugging level.

9. Select Fetches alerts to start ingesting issues.

   For all integrations, we recommend only fetching issues when everything is set up. When enabled, Cortex XSIAM searches for events that occurred within the time frame set for the integration, which is based on the specific integration. The default is 5 issues per minute.

   > **NOTE:**
   >
   > In some integrations, a classifier, an issue type, and mapper fields are included.

10. Test and Save & Exit.

1.8.3 | Configure integration permissions

Abstract

Integration permissions enable you to restrict running commands to specific roles in integrations.

You can use role-based access control (RBAC) to restrict running commands to specific roles at the integration instance level. If you have multiple instances of the same integration, you can assign different roles (permission levels) for the same command in each instance.

For example, you may want limit the roles that can run potentially harmful commands, such as the ability to isolate endpoints.

Users who do not have permission to run a command cannot do the following:

- Run the command from the CLI.
- Complete pending tasks in a Work Plan that uses the restricted command.
- Edit arguments for playbook tasks that use the restricted command.
- Select the command when editing a playbook.
- Leverage the restricted command when executing a reputation command, such as IP, Domain, and File.

If you have multiple instances of the same integration, you can assign different roles (permission levels) for the same command in each instance.

To view or edit integration permissions:

1. Go to Settings → Configurations → Data Collection → Integration Permissions.

   You can see a list of all enabled integrations.

2. Select the integration.

   You can see the following:

   - INSTANCE: Lists all instances for the integration.
   - COMMANDS: Lists all commands for the integration.
   - PERMITTED ROLES: Lists the roles that have permission to run the command. Default is No Restrictions.

3. For a specific command, restrict the roles that can run the command.

   1. Go to the relevant command.
   2. Click Edit.
   3. In the PERMITTED ROLES, column, select the roles that you want to allow running the command.

4. Save the integration permissions.

1.8.4 | Fetch issues from an integration instance

Abstract

Configure a third-party integration instance to fetch issues into Cortex XSIAM cases for investigation.

You can poll third-party integration instances for events and turn them into Cortex XSIAM issues (fetching). Many integrations support fetching, but not all support this feature. You can view each integration in the Developer Hub.

When setting up an instance, you can configure the integration instance to fetch events. You can also set the interval for which to fetch new issues by configuring the Issue Fetch Interval field. The fetch interval default is 1 minute. This enables you to control the interval in which an integration instance reaches out to third-party platforms to fetch issues into Cortex XSIAM.

> **NOTE:**
>
> - In some integrations, the Issue Fetch interval is called Feed Fetch Interval.
> - If the integration instance does not have the Issue Fetch Interval field, you need to add this field by editing the integration settings. If the integration is from a content pack, you need to create a copy of the integration. Any future updates to this integration will not be applied to the copy integration.
> - If you turn off fetching for a while and then turn it on or disable the instance and enable it, the instance remembers the last run and pulls all events that occurred while it was off. If you don't want this to happen, verify that the instance is enabled and click Reset the “last run” timestamp when editing the instance. Also, note that "last run" is retained when an instance is renamed.

After configuring the instance, you may need to set up a correlation rule to generate issues in Cortex XSIAM.

Correlation rules are predefined logic or patterns that Cortex XSIAM uses to identify relationships between disparate events occurring across an organization's IT environment. If the conditions specified in the rule are met, Cortex XSIAM generates an issue.

How to fetch issues

1. Go to Settings → Configuration → Data Collection → Automation and Feed Integrations, find the integration, and click + Add instance.
2. In the integration's dialog box, select Fetch issues.

   After this setting is enabled, Cortex XSIAM searches for events that occurred within the time frame set for the integration, which is based on the specific integration. The default is 10 minutes, but it can be changed in the integration script.

3. (Optional) In the Issue Fetch Interval field, set the interval of hours and minutes to fetch alerts (default 1 minute).
4. (Optional) If the Issue Fetch Interval field does not appear, add it to the integration.

Relevant for any issue fetching integration:

a. For integrations installed from a content pack, select the duplicate integration button.

If you have already duplicated the integration, click the Edit integration's source button.

b. In the Basic section, select the Fetch issues checkbox.

In the Parameters section, you can see that the `IssueFetchInterval` parameter is added. Change the default value if necessary.

c. Click Save to save the changes.

5. To generate issues in Cortex XSIAM, add a correlation rule, as required.

> **NOTE:**
>
> Some content packs include preconfigured correlation rules, but you should review them to see if they suit your use case and duplicate them if required. Go to Threat Management → Detection Rules → Correlations, search for the relevant rule, right-click, and select Preview Rule. For example, the ServiceNow v2 Alerts (automatically generated) correlation rule uses the following XQL Query:
>
> ```
> dataset = servicenow_v2_generic_alert_raw
> | filter _alert_data != null
> | alter alert_severity = json_extract_scalar(_alert_data, "$.severity")
> | alter alert_category = json_extract_scalar(_alert_data, "$.alert_category")
> | alter alert_name = json_extract_scalar(_alert_data, "$.alert_name")
> | alter alert_description = json_extract_scalar(_alert_data, "$.alert_description")
> ```
>
> You may want to update the query by defining complex, multi-source detection logic or add filters, such as alert severity or assignee.

1.8.4.1 | Map fields to issue types

Abstract

You can create independent mappers for integrations.

Mappers enable you to map information from incoming events to the issue fields that you have in your system. You can map to system issue fields or custom issue fields.

Mapping event attributes or issue fields takes place in two stages. First you map all of the fields that are common to all issues in the default mapping. Second, you map the additional fields that are specific for each issue indicator type, or overwrite the mapping that you used in the default mapping.

> **NOTE:**
>
> In the Classification & Mapping page, the mapping does not indicate for which issue types they are configured. Therefore, when creating a mapper, it is best practice to add to the mapper name, the issue types the mapper is for. For example, Mail Listener - Phishing.

> **NOTE:**
>
> When mapping a list, we recommend you map to a multi select field. Short text fields do not support lists. If you do need to map a list to a short text field, add a transformer in the relevant playbook task, to split the data back into a list.

You can use this procedure for creating a classifier or duplicating an existing mapper for issue types.

1. Go to Settings → Configurations → Object Setup → Issues → Classification & Mapping.
2. Click New and select Issue Mapper (incoming). The Issue Mapper maps all of the fields you are pulling from the integrations to the issue fields in your layouts.
3. Under Get data, select from where you want to pull the information based on where you want to map the issue types.
   - Pull from instance - select an existing integration instance.
   - Select schema - when supported by the integration, this pulls all of the fields for the integration from the database. This enables you to see all of the fields for each given event type that the integration supports.
   - Upload JSON - upload a formatted JSON file which includes the field you want to map.
4. Under Issue Type, start by mapping out the Common Mapping. This mapping includes the fields that are common to all of the issue types and will save time having to define these fields individually in each issue type.
5. Click the event attribute to which you want to map. You can further manipulate the field using filters and transformers.

   You can click Auto Map to automatically map fields with common or similar names to fields in Cortex XSIAM . For example, Severity to Importance or Description to Description.
6. Repeat this process for the other issue types for which this mapping is relevant.
7. Click Save.
8. Go to Settings → Configurations → Data Collection → Automation & Feed Integrations.

   a. Select the integration instance to which you want to apply the mapper.

b. In the integration settings, under Mapper (incoming) select the mapper you created and click Save.

1.8.4.2 | Classify events using a classifier for issue types

Abstract

Classify events using a classification key in an integration ingestion.

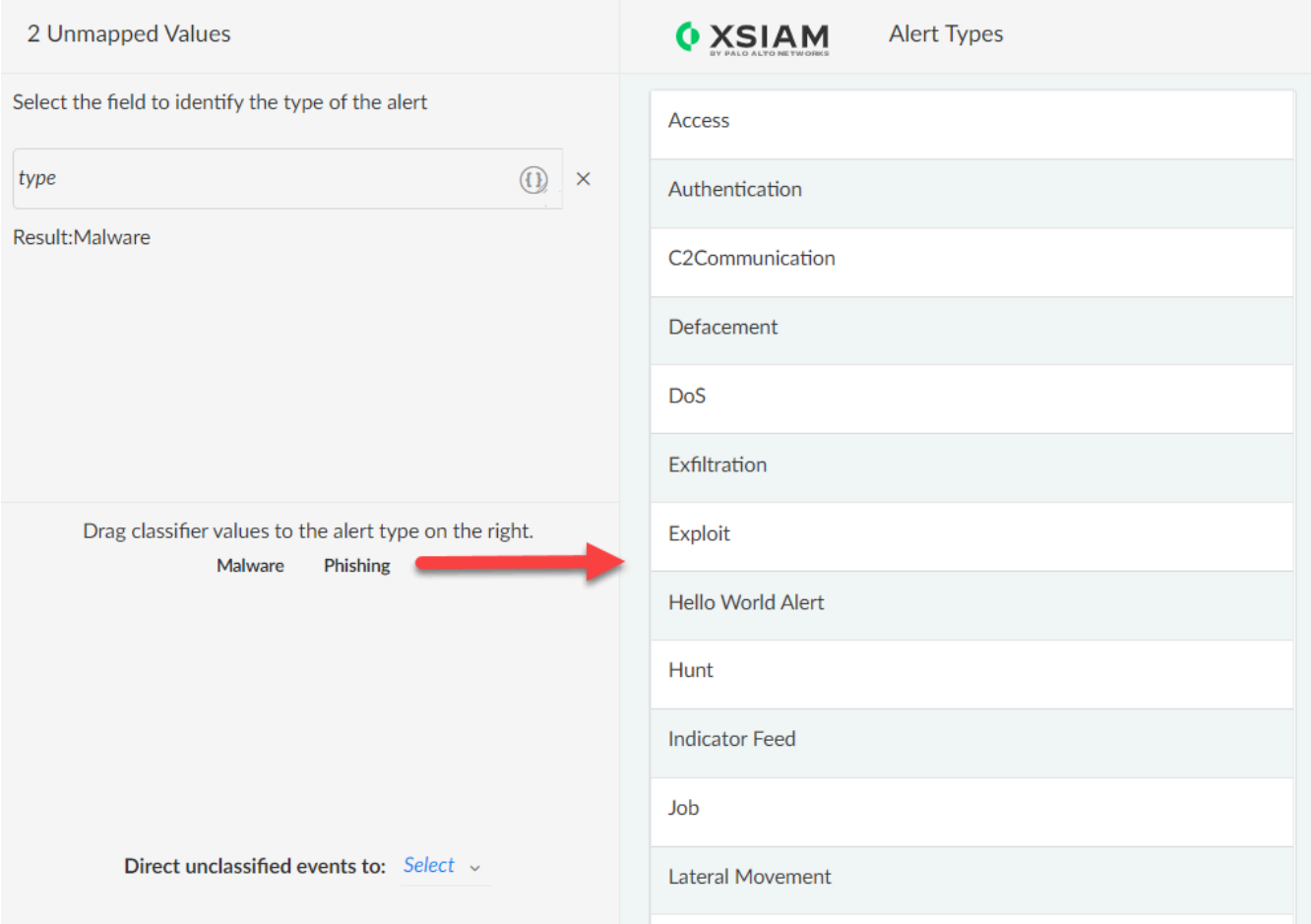
When an integration fetches issues, it populates the rawJSON object in the issue object. The rawJSON object contains all of the attributes for the event. For example, source, when the event was created, the priority that was designated by the integration, etc. When classifying the event, you want to select an attribute that can determine the event type.

You can use this procedure for creating a classifier or duplicating an existing classifier.

1. Go to Settings → Configurations → Object Setup → Issues → Classification & Mapping.
2. Click New and select Issue Classifier.

   If you want to duplicate the classifier, select the relevant classifier and then duplicate it.
3. Under Get data, select from where you want to pull the information based on which you will classify the issue types.
   - Pull from instance - select an existing integration instance.
   - Select schema - when supported by the integration, this will pull all the fields for the integration from the database from which you can select by which to classify the events.
   - Upload JSON - upload a formatted JSON file which includes the field by which you want to classify.
4. In the Select Instance field, select the instance from where you want to choose the value.
5. In the Data fetched from select the value by which you want to classify the events.
6. Drag values from the Unmapped Values column to the relevant issue type on the right.

   You can optionally choose a default issue type for unclassified issues from Direct unclassified events to: Select.



7. Click Save.

8. Go to Settings → Configurations → Data Collection → Automation & Feed Integrations.

    a. Select the integration to which you want to apply the classifier.

    b. In the integration settings, under Classifier, select the classifier you created and click Save.

1.8.5 | Troubleshoot Integrations

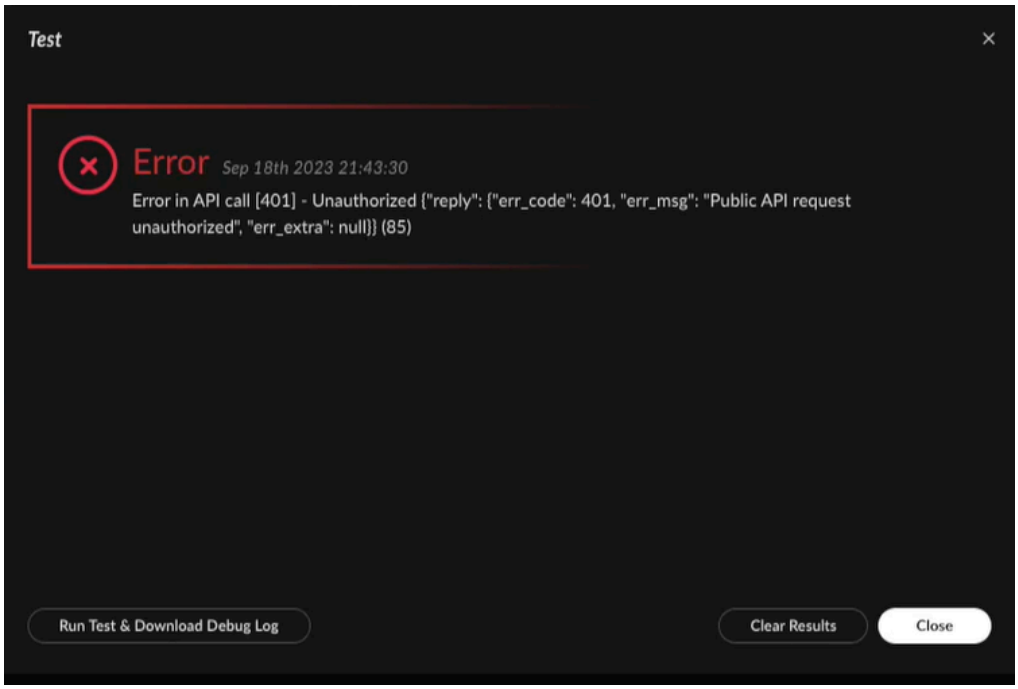
Abstract

Learn how to troubleshoot your integration in Cortex XSIAM.

When troubleshooting integrations, do the following:

- Use the Test button in the integration instance.
- Verify the integration settings. Check settings such as usernames, URLs, and passwords.
- Download the debug log file and review its contents.

  In the following example, you receive a 401 unauthorized error code after testing the integration.

  Click Run Test & Download Debug Log, to download the debug file locally. You can verify what server the URL request is being forwarded to and any other reasons as to why you received this error code. The 401 unauthorized error code usually relates to invalid error credentials, expired tokens, or incorrect API settings.

- Enable verbose or debug-level logging on the integration.

If you are unable to fix the integration, contact Customer Support for further assistance.

1.8.6 | Forward Requests to Long-Running Integrations

Abstract

Configure and manage long-running integrations to export internal data from Cortex XSIAM.

Some long-running integrations provide internal data via API calls to your third-party software, such as a firewall. You can set up Cortex XSIAM to allow third-party software to access long-running integrations installed either on the Cortex XSIAM tenant or on an engine.

Long-running integrations provide internal data via API calls such as:

Integration	Description	See More
O365 Teams (Using Graph API)	Get authorized access to a user's Teams app in a personal or organizational account.	O365 Teams (Using Graph API)
Generic Webhook	Creates cases on event triggers. The trigger can be any query posted to the integration.	Generic Webhook
Generic Export Indicators Service	Use the Generic Export Indicators Service integration to provide an endpoint with a list of indicators as a service for the system indicators. You can set up the tenant to export internal data to an endpoint. **NOTE:** This integration replaces the External Dynamic list integration, which is deprecated. For more information about how to set up the integration, see Manage external dynamic lists.	Generic Export Indicators
Microsoft Teams	Send messages and notifications to team members.	Microsoft Teams
TAXII Server	Provides TAXII Services for system indicators (Outbound feed).	TAXII Server
TAXII2 Server	Provides TAXII2 Services for system indicators (outbound feed). You can choose to use TAXII v2.0 or TAXII v2.1.	TAXII2 Server
PingCastle	Listens for PingCastle XML reports.	PingCastle
Publish List	Publishes Cortex XSIAM lists for external consumption.	Publish List
Simple API Proxy	Provides a simple API proxy to restrict privileges or minimize the number of credentials issued at the API.	Simple API Proxy
Syslog v2	Opens cases automatically from Syslog clients.	Syslog v2
Web File Repository	Make your environment ready for testing purposes for your playbooks or automations to download files from a web server.	Web File Repository

NOTE:

- When running on the tenant, you can only use long-running integrations provided by Cortex XSIAM, you cannot create custom ones. Custom long-running integrations are supported only on engines at this time.
- Configuring custom certificates or private API Keys in the long-running integration instance is supported only on engines, not on the Cortex XSIAM tenant.

Credentials

For long-running integrations running on a tenant, you must set a username and password. For long-running integrations running on an engine, we strongly recommend setting a username and password, but it is not required.

Users with sufficient permissions can set the username and password for specific integration instances on the Automation & Feed Integrations page.

Test the long-running integration connection

- Integration instance running on a tenant

  You can use CURL commands from any terminal to access and test the long-running integration. The string `xdr` in the URL must be replaced by `crtx` and the data URL must always be prefixed by `ext-`.

  > **NOTE:**
  >
  > For the TAXII Server and TAXII2 Server integrations, the `xdr` string is automatically replaced by `crtx`. For the Microsoft Teams integration, you can use the `microsoft-teams-create-messaging-endpoint` command to get the correct messaging endpoint based on the server URL, the server version, and the instance configurations. For more information, see Microsoft Teams.

  Example:

  Tenant URL: https://crtx-cnt-onr-xsiam-dran-9c0.xdr-qa2-uat.us.com

  Request URL: https://ext-crtx-cnt-onr-xsiam-dran-9c0.crtx-qa2-uat.us.com/xsoar/instance/execute/edl_instance_01\?q\=type:ip

  CURL: curl -v -u user:pass https://ext-crtx-cnt-onr-xsiam-dran-9c0.crtx-qa2-uat.us.com/xsoar/instance/execute/edl_instance_01\?q\=type:ip

- Integration instance running on an engine

  You can use CURL commands from any terminal to access and test the long-running integration at the engine URL:

  `http://<engine-address>:<integration listen port>/`

  For example, `curl -v -u user:pass http://<engine_address>:<listen_port>/?n=50`

Curl request parameters

When sending a curl request to the URL, use the following parameters:

Argument	Description	Example
`n`	The maximum number of entries in the output. If no value is provided, will use the value specified in the List Size parameter in the integration instance settings.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?n=50`
`s`	The starting entry index from which to export the indicators.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?s=10&n=50`
`v`	The output format. Supports PAN-OS (text), CSV, JSON, mwg, and proxysg (alias: bluecoat).	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=json`
`q`	The query is used to retrieve indicators from the system.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?q="type:ip and sourceBrand:my_source"`
`t`	Only with mwg format. The type is indicated at the top of the exported list. Supports: string, applcontrol, dimension, category, ip, mediatype, number, and regex.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=mwg&t=ip`
`sp`	If set, will strip ports off URLs; otherwise, will ignore URLs with ports.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=text&sp`
`di`	Only with PAN-OS (text) format. If set, will ignore URLs that are not compliant with PAN-OS URL format instead of being rewritten.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=text&di`

Argument	Description	Example
`cr`	If set, will strip protocols off URLs.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=text&pr`
`cd`	Only with proxysg format. The default category for the exported indicators.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=proxysg&cd=default_category`
`ca`	Only with proxysg format. The categories that will be exported. Indicators not in these categories will be classified as the default category.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=proxysg&ca=category1,category2`
`tr`	Only with PAN-OS (text) format. Whether to collapse IPs. • 0 - Do not collapse. • 1 - Collapse to ranges. • 2 - Collapse to CIDRs	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?q="type:ip and sourceBrand:my_source"&tr=1`
`tx`	Whether to output CSV formats as textual web pages.	`https://ext-<tenant-address>/instance/execute/<ExportIndicators_instance_name>?v=csv&tx`

Define a listening port for long-running integrations

When configuring a long-running integration instance, you may need to define a listening port.

- Integration Instance Running on a Tenant

  If the long-running integration runs on the Cortex XSIAM tenant, you do not need to enter a Listen Port in the instance settings. The system auto-selects an unused port for the long-running integration when the instance is saved.

- Integration Instance Running on an Engine

  You must set the Listen Port for access when configuring a long-running integration instance on an engine. Use a unique port for each long-running integration instance. Do not use the same port for multiple instances.

1.9 | Verify collector connectivity

Abstract

Verify collector connectivity and troubleshoot collector errors.

You can verify the connectivity status of a collector instance on the Data Sources page. Instances are grouped by integration, and a status icon shows a summary of instance statuses for each integration. Expand the integration section to see the status of each individual instance, and hover over the status icons to see details about warning or error statuses.

In addition, Cortex XSIAM creates Collection health issues if connectivity disruptions occur in your collection integrations, custom collectors, and Marketplace integrations. For more information, see About health issues.

Troubleshooting collector errors

> **NOTE:**
>
> For more information on troubleshooting data collector applet errors, see Troubleshoot Broker VM applet connectivity.

Where can I see if I have a connectivity error on a collector instance?

On the Data Sources page, instances in error status display an error icon. Hover over the error icon next to the instance name to see the error message as received from the API.

Where can I trace the connectivity changes of a collector instance?

Each status change of an instance is logged in the `collection_auditing` dataset. Querying this dataset can help you see all the connectivity changes of an instance over time, the escalation or recovery of the connectivity status, and the error, warning, and informational messages related to status changes.

Example 164.

This example searches for status changes on Strata IOT integrations:

```
dataset = collection_auditing
|filter collector_type = "STRATA_IOT"
```

How can I set up correlation rules to trigger collection issues?

Cortex XSIAM provides OOTB Collection issues that are triggered when a data collector instance is in error status, which means it is disconnected or not sending data. In addition, you can set up your own correlation rules that trigger collection issues for your specific needs. For example, you might want to be notified if a high-profile collector is in warning status so that you can fix the problem and prevent the collector from disconnecting.

Example 165. Example: Trigger collection issues for warning statuses on the STRATA_IOT collector

In this example, a correlation rule triggers a Collection issue if an integration of the Strata IOT collector changes to warning status. Any issues will appear on the Health Issues page.

Example XQL:

```
dataset = collection_auditing
|filter classification = "Warning" and collector_type = "STRATA_IOT"
```

Additional fields to specify in the correlation rule:

Field	Value
Time Schedule	Hourly
Query time frame	1 Hour
Issue Suppression	Select Enable issue suppression.
Action	Select Generate issue.
Issue Domain	Health
Severity	Medium
Category	Collection **NOTE:** If an issue is triggered, the investigation options in the right-click menu of the Health Issues pages are context-specific. Make sure that you specify the relevant issue category.

1.10 | Overview of data ingestion metrics

Abstract

Learn more about the data ingestion health metrics in the `metrics_source` dataset and the `metrics_view` preset.

PREREQUISITE:

For Cortex XSIAM to monitor data ingestion health and create health issues, you must enable the following settings under Configurations:

- Cortex - Analytics: Go to Configurations → Cortex - Analytics. For more information, see Enable the Analytics Engine and Identity Analytics.
- Data Ingestion Monitoring: Go to Configurations → General → Server Settings → Data Ingestion Monitoring. For more information, see Set up your environment.

The data ingestion metrics are calculated in 5-minute aggregation periods and saved to the `metrics_source` dataset and `metrics_view` preset. These metrics measure the amount, size, and rate at which logs are ingested by a data source:

Metric	Description
total_size_bytes	Total size (in bytes) of the logs collected during the aggregation period.
total_size_rate	Average size (in bytes per second) of the logs collected during the aggregation period.
total_event_count	Total number of logs collected during the aggregation period
total_event_rate	Average number (in count per second) of logs collected during the aggregation period.

In the `metrics_source` dataset, the data ingestion metrics are saved alongside additional fields that describe the data source associated with the metrics. Only entries with ingestion metric values greater than zero are saved in the dataset. Entries with zero values are not saved in this dataset.

`metrics_view` is a preset for data in the `metrics_source` dataset. The preset also simulates completion of entries with zero values in data ingestion metrics at runtime, which allows effective use of metrics. Therefore, when investigating disruptions in data collection, we recommend using the `metrics_view` preset in XQL queries and correlation rules.

Cortex XSIAM built-in data ingestion monitoring and issue mechanism uses the data ingestion metrics to identify disruptions in the data ingestion pipeline. Using analytical logic, Cortex XSIAM creates an ingestion baseline for each data source that reflects the routine pattern of log collection. If a data source isn't ingesting logs, or there is a significant deviation from the baseline, ingestion issues are triggered. You can see all ingestion issues on the Health Issues page. To troubleshoot or investigate an issue, right-click an issue and click Investigate in XQL query. For more information, see Investigate and resolve health issues.

In addition, you can create your own custom logic for data ingestion health monitoring by setting up correlation rules that monitor the data ingestion metrics. For more information, see Creating correlation rules to monitor data ingestion health.

The following table describes all the fields in the `metrics_source` dataset and `metrics_view` preset:

Read more...

Field	Type	Description
total_size_bytes	Integer	Total size (in bytes) of the logs collected during the aggregation period.
total_size_rate	Integer	Average size (in bytes per second) of the logs collected during the aggregation period.
total_event_count	Integer	Total number of logs collected during the aggregation period.
total_event_rate	Integer	Average number (in count per second) of logs collected during the aggregation period.
data_freshness_max_delay	Float	Maximum delay value from all log entries in a record between log creation at the source and ingestion into Cortex XSIAM (in seconds).
data_freshness_median	Float	Median delay value from all log entries in a record between log creation at the source and ingestion into Cortex XSIAM (in seconds).

Field	Type	Description
data_freshness_ninetieth_percentile	Float	Ninetieth percentile of delay values from all log entries in a record between log creation at the source and ingestion into Cortex XSIAM (in seconds).
last_seen	Datetime	Time that the last logs were collected.
_vendor	String	Vendor of the observing data source.
_product	String	Product name of the observing data source.
_device_id	String	(For firewall devices) Device ID
_log_type	String	(For firewall devices) Log type
_collector_type	String	(Event Metadata) Type of collector that provided the log.
_collector_name	String	(Event Metadata) Name of the collector instance.
_collector_id	String	(Event Metadata) ID of the XDR Collector.
_collector_ip	String	(Event Metadata) IP address of the XDR Collector.
_reporting_device_name	String	(Event Metadata) Host name of the device where the log originated.
_reporting_device_ip	String	(Event Metadata) IP Address of the device where the log originated.
_final_reporting_device_name	String	(Event Metadata) Hostname of the device that the log was extracted from.
_final_reporting_device_ip	String	(Event Metadata) IP of the device that the log was extracted from.
_broker_device_name	String	(Event Metadata) Host name of the Broker VM.
_broker_device_ip	String	(Event Metadata) IP address of the Broker VM.
_broker_device_id	String	(Event Metadata) ID of the Broker VM.
_time	Datetime	Timestamp of the interval.
_insert_timestamp	Datetime	Recorded time of the entry.

1.10.1 | Creating correlation rules to monitor data ingestion health

Abstract

See examples of correlation rules for monitoring data ingestion health.

In addition to the OOTB Ingestion health issues, you can build your monitoring logic for ingestion by creating correlation rules that are specific to your requirements. You can create rules that monitor the data ingestion metrics for a specific source within a specific timeframe, and trigger ingestion health issues if there is a deviation from the regular pattern of log collection.

The following examples can help you set up your own correlation rules with the data ingestion metrics:

Example 1: No logs collected from a data source for 1 hour

In this example, the correlation runs every hour and calculates the number of logs that are collected for each data source over the previous hour. If no logs are collected for a data source during an aggregation period, a security issue is triggered.

Example XQL:

```
preset = metrics_view
| comp sum(total_event_count) as total_event_count_sum by _collector_id, _collector_ip,
_collector_name, _collector_type, _final_reporting_device_ip, _final_reporting_device_name,
 _broker_device_id, _vendor, _product
| filter total_event_count_sum = 0
```

Addition fields to specify in the correlation rule:

Field	Value
Time Schedule	Hourly
Query time frame	1 Hour
Issue Suppression	Select Enable issue suppression.
Fields	Uncheck `total_event_rate_sum`, leave other fields checked.
Action	Select Generate issue.
Issue Domain	Health
Severity	High
Type	Ingestion
Issue Fields Mapping	Select Use preconfigured fields to map the fields that are relevant to data ingestion health.

Example 2: No logs received from a Firewall for 20 minutes

In this example, the correlation runs every 20 minutes and calculates the number of logs that are received for each firewall in a lookup dataset during the last 20 minutes. If no logs are received from a device during an aggregation period, a security issue is triggered.

Example XQL:

```
preset = metrics_view
| join conflict_strategy = left  type = inner (dataset = ngfw_device_Id_keepalive
| fields _device_id) as devices devices._device_id = _device_id  | comp sum(total_event_count)
 as total_event_count_sum by _device_id, _product,_vendor
| filter total_event_count_sum = 0
```

Addition fields to specify in the correlation rule:

Field	Value
Time Schedule	Every 20 minutes

Field	Value
Query time frame	20 minutes
Issue Suppression	Select Enable issue suppression.
Fields	Uncheck `total_event_rate_sum`, leave other fields checked.
Action	Select Generate issue.
Issue Domain	Health
Severity	High
Type	Collection
Issues Fields Mapping	Select Use preconfigured fields to map the fields that are relevant to data ingestion health.

1.10.2 | Measuring data freshness

Abstract

Learn more about the data freshness metrics collected by Cortex XSIAM.

Cortex XSIAM provides metrics that calculate the freshness of your ingested data and highlight delays in your data collection. The metrics calculate the freshness delay value by measuring the difference between log creation at the source (`_TIME`) and ingestion into Cortex XSIAM (`_INSERT_TIME`).

Metrics are collected and calculated per data source during five-minute aggregation periods and allocated into the following buckets. The recorded freshness delay value is the top value in the range of the bucket:

- 0 to 30 seconds → 30 seconds
- 30 to 60 seconds → 60 seconds
- 60 seconds to 5 minutes → 300 seconds
- 5 minutes to 1 hour → 3,600 seconds
- 1 hour to 24 hours→ 86,400 seconds
- 24 hours to week→ 604,800 seconds

Metric	Description
data_freshness_max_delay	Maximum freshness delay value among all log entries in an aggregation period. This reflects the worst case.
data_freshness_median	Median freshness delay value among all log entries in an aggregation period. 50% of values are smaller than the median, and 50% of values are higher or equal to the median.
data_freshness_ninetieth_percentile	Ninetieth percentile of delay values among all log entries in an aggregation period. This delay value is 90% higher than other log entry differences. It reflects the worst case, but eliminates the spikes.

The metrics are saved to the `metrics_source` dataset and are also available in the `metrics_view` preset.

> **NOTE:**
>
> - The max_delay metric is taken from the maximum bucket value with a restricted limit; therefore, metrics show whole numbers.
> - The median and ninetieth_percentile metrics are statistical calculations that give an approximation of the real value; therefore, metrics show decimal numbers.
> - Time slots with a zero log count or zero byte count display records with zero values. Subsequently, the data freshness metrics will also have zero values.
> - Timezone differences between `_TIME` and `_INSERT_TIME` might cause time skews with negative differences. Negative differences are rounded to zero values.

1.11 | About health issues

Abstract

Cortex XSIAM provides health issues to help you monitor the health and integrity of supported Cortex XSIAM resources. Health issues comprise ingestion, collection, correlation, and event forwarding errors.

> **PREREQUISITE:**
>
> For Cortex XSIAM to monitor data ingestion health and create health issues, you must enable the following settings under Configurations:
>
> - Cortex - Analytics: Go to Configurations → Cortex - Analytics. For more information, see Enable the Analytics Engine and Identity Analytics.
> - Data Ingestion Monitoring: Go to Configurations → General → Server Settings → Data Ingestion Monitoring. For more information, see Set up your environment.

Cortex XSIAM provides health issues to help you monitor the health and integrity of supported Cortex XSIAM resources. Health issues provide insights into health drifts, such as failure events or status changes. The issues help you stay on top of your health related errors and ensure optimal performance in Cortex XSIAM. In addition, you can set up notifications on health issues.

Health issues are associated with the Health Domain. When setting up notification forwarding or other configurations for health issues, use the filter Issue Domain = Health.

To view health issues, go to Settings → Health Issues, or on the Issues page select the Health Domain table view. Click an issue to see more details in the issue card, or right-click to take actions and investigate an issue. For more information, see Investigate and resolve health issues.

> **NOTE:**
>
> The Health Issues page displays issues that were triggered after July 2024. To see health issues that were triggered before this date, click Legacy Health Issues.

Types of health issues

Cortex XSIAM provides the following types of OOTB health issues:

- Ingestion issues: Triggered by interruptions in data ingestion, or deviation from the calculated ingestion baseline
- Collection issues:: Triggered by connectivity errors in your collection integrations, custom collectors, and Marketplace integrations
- Correlation issues: Triggered by correlation rules that complete with an error status

> **NOTE:**
>
> Cortex XSIAM enforces the dedup logic to health issues. This logic reduces the likelihood of identical health issues from flooding the issues dataset.

Query health issue data

Health issues are associated with the Health domain. To query health issue data, use the following XQL:

```
dataset = alerts | filter alert_domain = "DOMAIN_HEALTH"
```

Health issue field descriptions

The following table describes the health issue fields.

Field	Description
Issue ID	A unique identifier that Cortex XSIAM assigns to each issue.

Field	Description
Issue Name	Name of the issue.
Issue Type	Type of health issue.
Issue Source	Source of the issue.
Broker VM ID	ID of the Broker VM.
Broker VM Name	Host name of the Broker VM.
Broker VM IP	IP address of the Broker VM.
Collector Name	Name of the collector instance.
Collector Type	Type of the collector.
Description	Text summary of the event including the issue source, issue name, and severity.
Device ID	Firewall device ID.
Excluded	Whether the issue is excluded.
External ID	Issue ID as recorded in the detector from which this issue was sent.
Final Reporting Device IP	IP of the device from which the log was extracted.
Final Reporting Device Name	Hostname of the device from which the log was extracted.
Ingestion Failure Duration	Amount of time that logs were not received or a drop in log ingestion was detected in minutes.
Observation Time	Time that the issue was observed in the system.
Playbook	Playbook that was run.
Playbook run status	Status of the playbook.
Product	Product name of the observing data source.

Field	Description
Resolution Status	Status that was assigned to this issue when it was triggered (or modified). Right-click an issue to change the status. If you set the status to Resolved, select a resolution reason.
Reporting Device Name	Host name of the device where the log originated.
Reporting Device IP	IP Address of the device where the log originated.
Severity	Severity level that was assigned to this issue when it was triggered (or modified).
Starred	Whether the issue is starred by starring configuration.
Vendor	Vendor of the observing data source.
XDR Collector ID	ID of the XDR Collector.
XDR Collector IP	IP address of the XDR Collector.
XDR Collector Name	Host name of the XDR Collector.

1.11.1 | Investigate and resolve health issues

Abstract

You can investigate and take action on health issues from the Health Issues page and the Issues Table.

The following tasks explain how to investigate and resolve health issues. You can see health issues on the following pages:

- Go to Settings → Health Issues
- Go to Cases & Issues → Issues and change the table view to Health Domain.

Investigate data ingestion errors

A data ingestion issue identifies disruption in the data ingestion pipeline. For example, a data source is not sending logs, or there is a significant drop in log collection compared to the calculated ingestion baseline.

1. Identify the error: Type = Ingestion.
2. Right-click and select Investigate in XQL query.

   The Query Builder opens and runs a prefilled query to display related data ingestion metrics entries.
3. Review the query results.

   The results provide context for the issue and the events leading up to it. For more information about data ingestion metrics and setting up correlation rules with your own data ingestion logic, see Monitor data ingestion health.
4. Investigate data collector errors. Return to the Health Issues page, right-click the issue, and select Pivot to views → View collector details.

   Depending on the type of collector in error, the relevant data collector settings page opens, filtered by data collector.

Investigate collection errors

A collection issue identifies connectivity disruption in your collection integrations, custom collectors, and Marketplace integrations.

1. Identify the error: Type = Collection.
2. See the current status of the collector.

Right-click and select Pivot to views → View collector details. Depending on the type of collector in error, the relevant data collector settings page opens, filtered by data collector.

If the data collector is still in error, you can update the collector settings as required.

3. Investigate the collector error status.

   Run a query on the `collection_auditing` dataset to see all the connectivity changes of the collector over time, the escalation or recovery of the connectivity status, and the error, warning, and informational messages related to status changes.

   Example 166.

   This example searches for status changes for the "instance1" data collector integration:

   ```
   dataset = collection_auditing
   |filter collector_type = "STRATA_IOT" and instance = "instance1"
   ```

   For more information about troubleshooting collector errors and setting up correlation rules to trigger additional collection issues, see Verify collector connectivity.

Investigate correlation errors

A correlation issue identifies errors in your correlation rules.

1. Identify the error: Type = Correlation.
2. Right-click and select Investigate Correlation Auditing.

   The Query Builder opens and runs a prefilled query to display related correlation execution records.
3. Review the query results.

   Identify the correlation rule in error and take steps to resolve the error. For more information about how Cortex XSIAM identifies correlation rule errors, see Monitor correlation rules.

1.11.2 | Monitor data ingestion health

Abstract

Learn more about data ingestion health monitoring.

> **PREREQUISITE:**
>
> For Cortex XSIAM to monitor data ingestion health and create health issues, you must enable Data Ingestion Monitoring in your Server Settings. For more information, see Set up your environment.

Cortex XSIAM collects granular data ingestion metrics that provide an insight into the data ingestion pipeline, and identify disruptions in data collection. With these metrics you can trace data collection from a specific source, and see a breakdown by data source attributes such as Collector Name and Final Reporting Device.

You can use these metrics in Cortex Query Language (XQL) queries to investigate disruption and degradation in log collection. You can also create correlation rules that use your own data ingestion logic to trigger issues when disruption occurs for a specific data source within a specific timeframe.

In addition, Cortex XSIAM has a built-in data ingestion monitoring and issues mechanism that monitors the availability and overall health of data ingestion in your environment, and triggers ingestion health issues if disruptions occur.

Related topics

- Overview of data ingestion metrics
- Creating correlation rules to monitor data ingestion health
- Measuring data freshness
- About health issues

1.11.3 | Monitor correlation rules

Abstract

You can monitor your correlation executions with the `correlations_auditing` dataset.

Cortex XSIAM audits all correlation executions in the `correlations_auditing` dataset. The dataset records the query initiation times, end times, retry attempts, failure reasons, and other useful metrics. You can use this dataset to monitor your correlation executions. Cortex XSIAM also provides OOTB health issues that are generated when a correlation rule completes with errors. For more information, see About health issues.

In the `correlations_auditing` dataset, audit entries are added as follows:

- The rule starts executing. This is audited with the status of Initiated or Initiated Manually.
- The rule completes successfully. This is audited as Completed.
- The rule completes with errors. This is audited as Error.

> **NOTE:**
>
> In the dataset, the Query start time and Query end time indicate the timeframe of the data that was queried. The actual start and end times of the correlation rule execution are recorded in the _time field for the Initiated and Completed entries.

Field descriptions for the correlations_auditing dataset

The following table describes the fields in the correlations_auditing dataset:

Field	Description
_time	Timestamp of the audit. For entries with an Initiated or Initiated Manually status, this is the start time of the correlation rule execution. For entries with a Completed or Error status, this is the end time of the rule execution.
_id	Unique identifier of the audit entry.
Rule ID	Unique identification number for the correlation rule.
Name	Correlation rule name.
Status	The status of the correlation rule query. Possible values are Initiated, Initiated Manually, Completed, and Error.
Query start time	The start time of the query timeframe.
Query end time	The end time of the query timeframe.
Time frame	Time frame for the query.
Failure reason	For correlation rules with errors, this field displays the error message.
Retry attempts	Number of retry attempts before the query initiated or failed to run.
Schedule	Scheduled frequency to execute the correlation rule.
Rule creation time	Date and time that the correlation rule was created.
Rule modification time	Date and time that the correlation rule was last modified.

Field	Description
Description	Description of the correlation rule.
Severity	Defined severity of the correlation rule.
Dataset	Target data set, as defined in the correlation rule
Suppression status	Whether issue suppression is Enabled or Disabled.
Suppression duration	Duration for which to ignore additional events that match the issue suppression criteria.
Suppression fields	Fields on which the issue suppression is based.
Timezone	Timezone on which the scheduled frequency is based.
MITRE ATT&CK Tactic	MITRE ATT&CK tactic that the correlation rule attempted to generate.
MITRE ATT&CK Technique	MITRE ATT&CK technique that the correlation rule attempted to generate.
Issue category	Category of issue as configured when creating the rule.
Source	Source of the correlation rule.
XQL search	XQL query for the correlation rule.
Drill-down query	XQL query configured for further investigation.
Issue name	Name of the issue that the correlation rule will generate.