Lookout® Mobile Endpoint Security

SIEM Connector Guide

April 2018

Lookout

Copyright and disclaimer

Table of contents

Preface

Lookout Mobile Endpoint Security (MES) provides comprehensive risk management across iOS and Android devices to secure against app, device, and network-based threats while providing visibility and control over data leakage. With a seamless integration to your EMM solution, Lookout empowers your organization to adopt secure mobility without compromising productivity.

About this guide

This guide describes how to add the Lookout SIEM Connector to your mobile threat defense environment. Doing so enables Lookout mobile threat and device events to flow into your SIEM (Security Information and Event Management) environment.

Audience

This guide is for administrators, business users, and mobile security engineers who administer and support Lookout in their SIEM environment.

Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
User interface elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and field labels.
`Code sample`	This formatting is used for sample code segments within a paragraph.
<*Variable*>	This formatting is used for variable values. For variables within a code sample the formatting is `<Variable>`.
`File/path`	This formatting is used for filenames and paths.
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface, e.g., **File > New > Tag**.

Introduction

Installing the Lookout SIEM Connector in your computing environment enables Lookout mobile threat and device events to flow into your SIEM instance. This enables the integration of mobile security threat and device events into your company's security incident and event programs, as well as policies for action.

The Lookout SIEM Connector polls the Lookout RESTful Mobile Risk API every 30 seconds asking for new threat and device events that have occurred since the last successful poll. The app then writes the events into either the customer-configured syslog or directly to the receiver. You can then configure the various SIEM applications to obtain those events.

The Lookout SIEM Connector is a Docker image, compressed in TAR format, that you install into your environment.

Requirements

The Lookout SIEM Connector requires the following:

- 1 VM
- 2 GHz CPU
- 2 GB RAM
- 10 GB disk (~250 MB for the Docker image with JAR file and the remaining space for logging) If you plan to log with DEBUG information included, then you should allocate more disk space.

See the Lookout Mobile Endpoint Security Supported Platforms document for additional platform information.

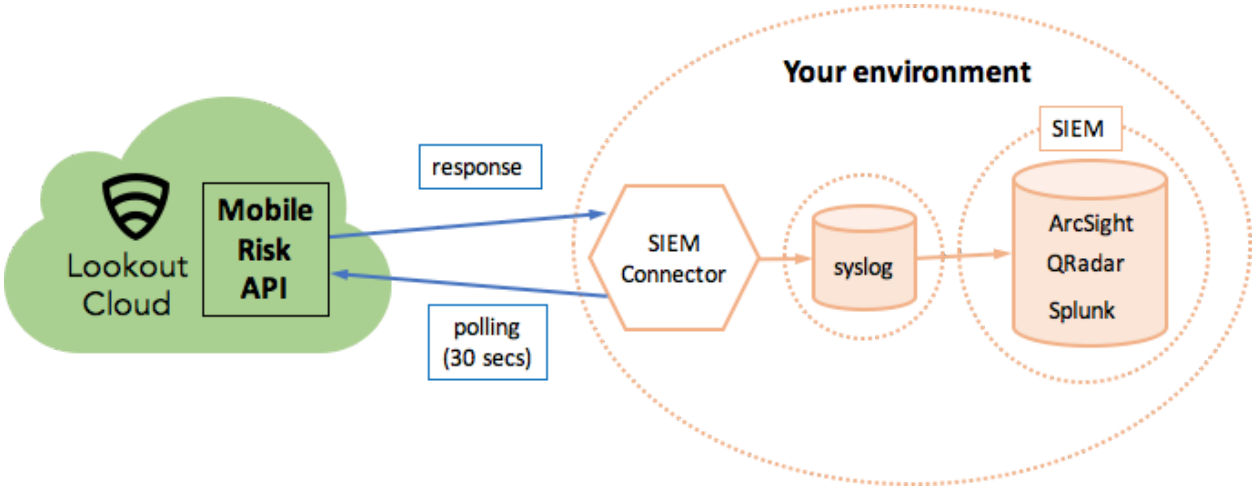
SIEM application configuration

The Lookout SIEM Connector supports these SIEM applications: **ArcSight, QRadar, Splunk**.

You can configure ArcSight, QRadar, or Splunk using either a syslog receiver or a socket receiver.

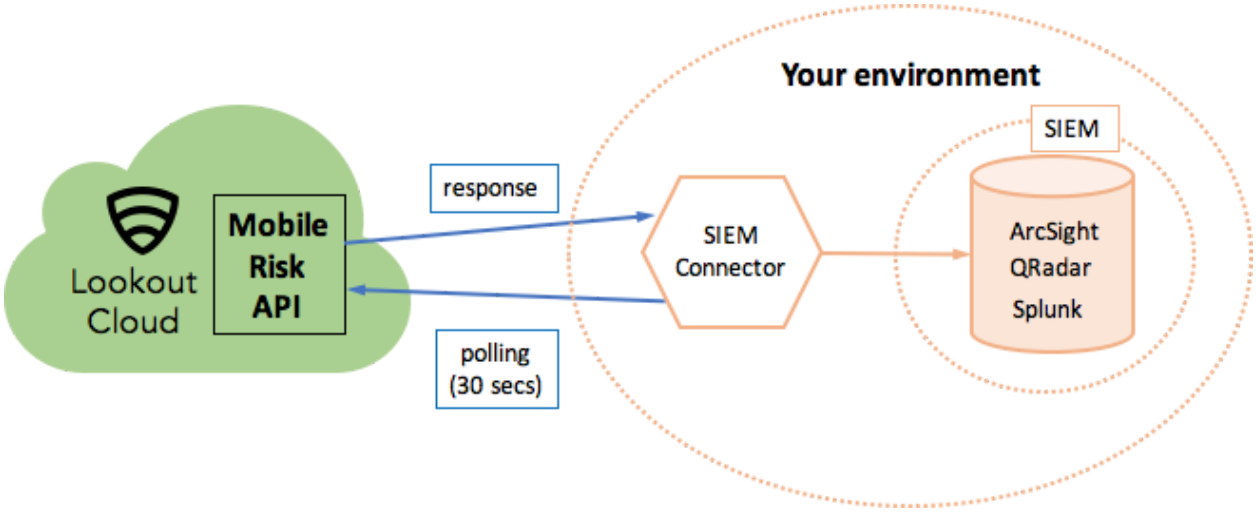
Syslog receiver

In this configuration the Lookout SIEM Connector writes to a syslog server and the SIEM application then reads from the syslog.



Socket receiver

In this configuration the Lookout SIEM Connector writes to the SIEM application's receiver directly.



Installation

Before you begin

Before you install the Lookout SIEM Connector verify that your SIEM application configuration is done properly, e.g., syslog/receiver is configured correctly and the ports are open. You'll need the following information about the destination of Lookout SIEM Connector events before you begin:

- Hostname or IP address
- Port
- Protocol
    - If SSL: keystore and trusted storage locations, passwords
    - If proxy: proxy hostname/IP, port, optional username/password

Docker installation

Docker is a free tool for creating, deploying, and running applications using containers. It is distributed under the Apache License 2.0.

Install Docker for your environment's operating system. Follow Docker's installation instructions from:

https://docs.docker.com/engine/installation/

> **NOTE:** The Lookout SIEM Connector image is built with Docker version: 1.12.2. However, you can install a newer Docker version since the connector is upward compatible.

Installing the Lookout SIEM Connector

If you're using QRadar, see the QRadar installation prerequisite section before starting your installation.

1. Get the latest version of the Lookout SIEM Connector from the Lookout Enterprise Support portal's SIEM Connector download area.
2. Click **Lookout SIEM Connector** to download the connector to the machine where you installed Docker.
3. Load and tag the Docker image.
    a. Load the Docker image

```
$ docker load --input siem-connector-<version>.tar
```

Example:

```
$ docker load --input siem-connector-0.1.7-63.tar
      7cbcbac42c44: Loading layer  5.05 MB/5.05 MB
      675e192b8244: Loading layer 122.3 MB/122.3 MB
      aba29c42eb15: Loading layer 20.52 MB/20.52 MB
      Loaded image ID:
```

```
Sha256:35f2d3578aa9cd8fc15a6d8e9613e5986987dde0a3497cdddede9693b9a9286c
```

b. Tag the Docker image.

View the Docker image and assign a tag to it. In this example you assign the tag of `lookoutsiem`. The tag requires the Image ID outputted from the images command as one of the inputs.

```
$ docker images
      REPOSITORY    TAG         IMAGE ID            CREATED             SIZE
      <none>        <none>      35f2d3578aa9        12 hours ago        145.2
MB
$ docker tag 35f2d3578aa9 lookoutsiem
$ docker images
      REPOSITORY    TAG         IMAGE ID            CREATED             SIZE
      lookoutsiem   latest      35f2d3578aa9        12 hours ago        145.2
MB
```

4. Create the configuration files.

a. Create a directory to store the configuration files. Your directory location can vary from this example:

```
$ mkdir -p /opt/LookoutSIEM/configs
```

b. Create the `application_key` file in the `/configs` directory

From the MES Console you can obtain a generated application key. See the Application key file section for the steps to obtain the application key. Create the `application_key` file using that generated key.

c. Create the `app.properties` file in the `/configs` directory.

See the Application properties file section for the fields you'll need in this file.

5. Run the Docker image in a container and mount the `/configs` directory you previously created.

NOTE: Consider adding this to a startup script so that the Docker image always runs at boot time. See the Docker documentation for details on using the `--restart always` flag when restarting containers.

```
$ docker run -d --name <container_name> -v <config_directory>:/conf
<image_tag>
```

In this example we use the container name `lookout`, the config directory `/opt/LookoutSIEM/configs`, and the image tag `lookoutsiem`.

```
$ docker run -d --name lookout -v /opt/LookoutSIEM/configs:/conf
lookoutsiem
```

To see the your `lookout` container running use:

```
$ docker ps -a
CONTAINER ID IMAGE        COMMAND                 CREATED       STAT       PORTS
NAMES
1c811c75d2c6 lookoutsiem  "sh -c 'java $JAVA_OP" 5 seconds ago Up 4 seconds
lookout
```

Updating the Lookout SIEM Connector

1. Get the latest version of the Lookout SIEM Connector from the Lookout Enterprise Support portal's SIEM Connector download area.
2. Click **Lookout SIEM Connector** to download the connector to the machine where you installed Docker.
3. Load and tag the Docker image.
    a. Load the Docker image:

    ```
    $ docker load --input siem-connector-<version>.tar
    ```

    b. Get the image ID for the new version of the connector (highlighted in the example below):

    ```
    $ docker images
    REPOSITORY                          TAG         IMAGE ID
    <host>/lookout/siem-connector       0.2.1-69    39b2d0e4b759
    ```

    c. Using the image ID, tag the new image with the `lookoutsiem` tag:

    ```
    $ docker tag 39b2d0e4b759 lookoutsiem
    ```

    This removes the tag from the previous version.

    d. Confirm the new image ID has the latest tag applied to the new `lookoutsiem` image:

    ```
    REPOSITORY                          TAG         IMAGE ID
    <host>/lookout/siem-connector       0.2.1-69    39b2d0e4b759
    lookoutsiem                         latest      39b2d0e4b759
    <host>/lookout/siem-connector       0.1.9-67    1c31132b5b0c
    ```

    e. List all running containers:

    ```
    $ docker ps -a
    CONTAINER ID    IMAGE           ...    NAMES
    281e5ab1bf9c    1c31132b5b0c    ...    lookout
    35927f4aae90    32d34d8a93c9    ...    lookout-v0.1.9-67
    ```

f. Stop the existing `lookout` container:

```
$ docker stop lookout
```

g. Rename the existing `lookout` container with the version it was running:

```
$ docker rename lookout lookout-v0.1.9-67
```

h. Start a new `lookout` container running the latest image:

```
$ docker run -d --restart always --name lookout -v
/opt/LookoutSIEM/configs:/conf lookoutsiem
```

i. Confirm that the new image is running and the previous version image is stopped:

```
$ docker ps -a
...  IMAGE          ...  STATUS                       NAMES
...  lookoutsiem    ...  Up 7 seconds                 lookout
...  1c31132b5b0c   ...  Exited (137) 45 seconds ago  lookout-v0.1.9-67
```

Monitoring and troubleshooting

Locate the Docker logs in their respective container at `/logs/LOapp.log`. To retrieve the tail of a container's logs, use:

`docker logs --follow <containerID` or `containerNAME>`

The most common errors for starting the application are firewall issues or a misconfigured `app.properties` file. Check to see if the application is running using the `docker ps -a` command.

Configuration files

You'll use two configuration files, `app.properties` and `application_key`, to define the Lookout SIEM connector. Any modifications to these files require a restart of your Docker container. The configuration files are mounted to the Docker container in its `/conf` folder. You can restart your container using:

```
docker restart <containerNAME>
```

Application properties file

Your `app.properties` file must contain values for all properties shown as required. Others are optional and you'll use them based on your deployment configuration and status, e.g. testing, production, etc.

Property name	Required?	Description
`connector.siemApp`	Yes	One of [`splunk, arcsight, qradar`]. Specifies the format used to map Lookout threat and device events.
`connector.syslog.host`	Yes	Specifies the syslog or SIEM connection hostname or IP address.
`connector.syslog.port`	Yes	Specifies the syslog or SIEM port.
`connector.syslog.protocol`	Yes	One of [`udp, tcp, ssl`]. Specifies the syslog or SIEM protocol to use.
`connector.ent.name`	No	Distinguishes which events come from which Lookout tenant. It is useful to filter events by tenants.
`connector.proxy.url`	No	When using a proxy, specifies the proxy URL used to connect to the Lookout Mobile Risk API, e.g., `http://proxy.com:80`
`connector.proxy.username`	No	When using a proxy, specifies the proxy username.
`connector.proxy.password`	No	When using a proxy, specifies the proxy user password.
`connector.logLevel`	No	One of [`info, debug, error`]. Specifies the logging level to use. The default is `info`. Change the default only when debugging. Set to `info` when running in production.

		NOTE: If you set the log level to debug, ensure you have sufficient disk space for the larger filesize.
`connector.logDirectory`	No	Specifies the directory for the log file. The default is `./logs`.
`connector.logFileName`	No	Specifies the log file name. The default is `LOapp.log`.
`connector.sslConfiguration.keyStore.path`	No (Yes for SSL)	Required when `connector.syslog.protocol=ssl`. The path to KeyStore for an SSL connection with syslog.
`connector.sslConfiguration.keyStore.password`	No (Yes for SSL)	Required when `connector.syslog.protocol=ssl`. The password to KeyStore for an SSL connection with syslog.
`connector.sslConfiguration.trustedStore.path`	No (Yes for SSL)	Required when `connector.syslog.protocol=ssl`. The path to TrustedStore for an SSL connection with syslog.
`connector.sslConfiguration.trustedStore.password`	No (Yes for SSL)	Required when `connector.syslog.protocol=ssl`. The password to TrustedStore for an SSL connection with syslog.
`connector.qradar.logSourceIdentifier`	QRadar only	Required for QRadar only when `connector.siemApp=qradar`. Specifies the IP address of the SIEM connector, must match the same as configured in QRadar log source.

A basic `app.properties` file may look like:

```
connector.siemApp=splunk
connector.syslog.host=192.168.99.100
connector.syslog.port=2514
connector.syslog.protocol=tcp
connector.ent.name=YourCompanyName
```

Application key file

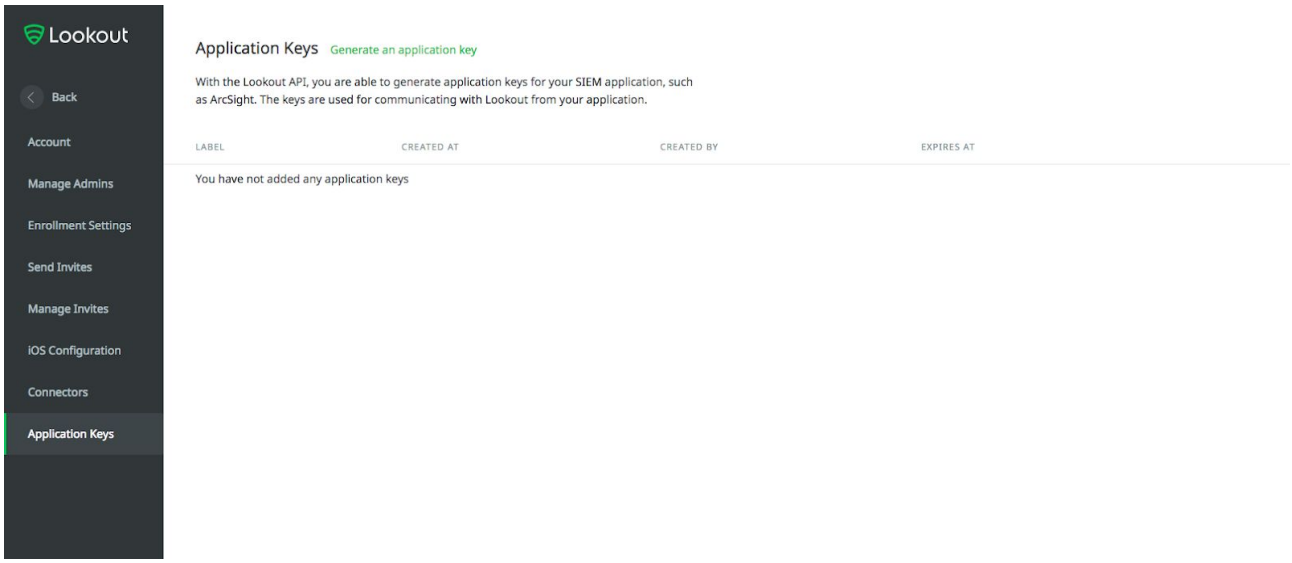
The Lookout SIEM Connector authenticates with the Lookout Mobile Risk API using OAuth 2.0. You need to have an application key specific to your Lookout tenant to properly configure your SIEM environment.

You store your key value in the Lookout Mobile Risk API `application_key` file in your Docker container.

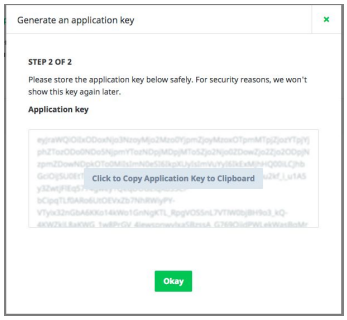
To retrieve your application key:

1. Log into the Lookout MES Console as an administrator.
2. Select **System > Application Keys**.

   NOTE: If you don't see the **Application Keys** tab, contact Lookout Enterprise Support to enable this feature for you.



3. Click **Generate an application key** and specify your label name.
4. Capture your generated key by clicking **Click to Copy Application Key to Clipboard**. This copies the key to your clipboard. This key is unique for the data in your Lookout tenant.



5. Copy the key from your clipboard into the `application_key` file on your Docker container.

   IMPORTANT: Copy the generated key to the `application_key` file immediately as you will not be able to see the key again after this procedure.

QRadar installation prerequisite

QRadar SIEM requires additional configuration prior to the Lookout SIEM Connector installation.

1. Download `lookout_qradar_config.xml` from the Lookout Enterprise Support portal SIEM Connector download area. This file contains QIDs, DSM IDs, and custom properties that allow QRadar to recognize Lookout events correctly.

2. Copy this file to your QRadar instance and import it using the Content Management script:

   ```
   /opt/qradar/bin/contentManagement.pl -a import -f
   <path>/lookout_qradar_config.xml
   ```

   Verify that the import was successful. You should see a QRadar message similar to:

   ```
   [INFO] Initializing Content Management Tool...
   [INFO] (ContentManagementCLI) Start Time: 2018-01-17 14:04:35
   [INFO] Starting import process
   [INFO] Summary of content found in bundle:
   [INFO]     Content Type - [Number of items]
   [INFO]         - dsmevent - [7]
   [INFO]         - ariel_property_expression - [48]
   [INFO]         - ariel_regex_property - [32]
   [INFO]         - qidmap - [7]
   [INFO] Summary of import/update operation:
   [INFO]     Content Type - [Number of items]
   [INFO]     Imported/Updated Content Summary -
   [INFO]         - ariel_regex_property    - [32]
   [INFO]         - ariel_property_expression    - [48]
   [INFO]         - qidmap    - [1]
   [INFO]         - dsmevent    - [7]
   [INFO]     Skipped Content Summary -
   ```

```
[INFO]          - qidmap     - [6]
[INFO]      Failed Content Summary -
[INFO]          -- No content failed
[INFO] Reloading sytem components, please wait
[INFO] Sending component reload notification
[INFO] SUCCESS: The import completed successfully. Please allow several minutes
for components to finish reloading.
```

For more QRadar information refer to the following IBM documentation:

https://www.ibm.com/support/knowledgecenter/SS42VS_7.2.8/com.ibm.qradar.doc/t_cmt_importing_content.html

3. Enter the IP of the **Log Source Identifier** as the Docker container's IP.
   See this IBM documentation to configure it:

   https://www.ibm.com/support/knowledgecenter/SSCQGF_7.2.0.1/com.ibm.IBMDI.doc_7.2.0.1/rg_conn_qradar_log_source.html

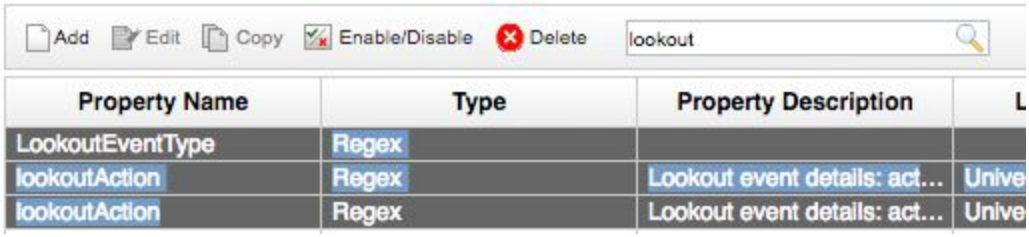
4. Add an additional property to the Lookout SIEM Connector `app.properties` file with the IP address from the previous step. See the Application properties file section for details.

   `connector.qradar.logSourceIdentifier=<IP address>`

Updating QRadar configuration

Before uploading a new version of the `lookout_qradar_config.xml` file, you must delete the existing lookout mappings.

1. Delete the existing mappings:
   a. In QRadar, select the **Admin** tab.
   b. In the sidebar, go to **Data Sources > Events** and click **Custom Event Properties** in the Events pane.
   c. In the **Search Properties...** field, enter `lookout`
   d. Multiselect all of the resulting fields that begin with "lookout" or "Lookout" and click **Delete**:



2. Upload the new mapping file as documented in the QRadar installation prerequisite topic.

Event fields

This section describes the contents of mobile threat and device events generated by the Lookout Mobile Risk API. There is a section specific to the field mappings of each supported SIEM; **ArcSight, QRadar,** and **Splunk**.

> **NOTE:** You can enable privacy controls for your Lookout tenant to prevent certain event field data from being mapped into your SIEM. An asterisk (*) denotes fields that you can make private. Contact Lookout Enterprise Support to enable privacy for your Lookout tenant.

ArcSight (CEF) fields

This section shows the mapping of Lookout event fields into ArcSight (CEF) fields.

Common fields

These fields are present in all Lookout events.

Lookout event field	Type	ArcSight (CEF) field	Description
n/a	String	`deviceVendor`	Static value of “Lookout” supplied.
n/a	String	`deviceProduct`	Static value of “SIEM Client” supplied.
n/a	String	`deviceVersion`	*<Lookout SIEM client version>*
n/a	String	`cs1`	Custom field `entName` with value *<entity name>* for events identification
`event.type`	Event type	`name`	Indicates the event type as [`THREAT`, `DEVICE`].
`event.id`	String	`devicePayloadId`	Unique ID of the event.
`event.eventTime`	DateTime	`rt`	Date and time of the event in the format `MMM dd yyyy HH:mm:ss.SSS` **NOTE:** Populating `rt` also populates `start` and `end` with the same value.

Threat event common fields

These fields exist in the details of Lookout events where `event.type=THREAT`.

Lookout event field	Type	ArcSight (CEF) field	Description
`event.details.id`	GUID	`deviceExternalId`	Lookout’s internal GUID for each threat.

event.details.type	Enum	deviceEventClassId	Indicates the threat event type as [APPLICATION,NETWORK, FILE,OS].
event.details.action	Enum	act	Indicates the state of the threat action as [DETECTED, RESOLVED, IGNORED].
event.details.severity	Enum	severity	Indicates the severity of the threat event as [LOW, MEDIUM, HIGH].
event.details.classifications	Array	cat	Indicates the classification type of the event. Can be any of [ROOT_ENABLER, RISKWARE, ADWARE, CHARGEWARE, DATA_LEAK, TROJAN, WORM, VIRUS, EXPLOIT, BACK_DOOR, BOT, APP_DROPPER, CLICK_FRAUD, SPAM, SPYWARE, SURVEILLANCEWARE, VULNERABILITY, ROOT_JAIL_BREAK, CONNECTIVITY, TOLL_FRAUD, SIDELOADED_APP, ACTIVE_MITM, UNKNOWN].

Network threat fields

These fields exist in the details of Lookout threat events where event.details.type=NETWORK.

Lookout event field	Type	ArcSight (CEF) field	Description
*event.details.ssid	String	cs2	Custom field networkSSID with value *<wireless network ID>.*
*event.details.macAddress	String	dvcmac	The MAC address.
event.details.proxyProtocol	String	app	The proxy protocol being used.
event.details.proxyPort	Integer	destinationTranslatedPort	The proxy port being used.
event.details.proxyAddress	String	destinationTranslatedAddress	The proxy IP address.
event.details.vpnLocalAddress	String	dvc	The VPN IP address.
event.details.vpnPresent	Integer	cn1	Custom field vpnPresent that Indicates if a VPN exists. 1 = True, 0 = False.
event.details.	String	flexString2	Custom field dnsIpAddresses that

dnsIpAddresses	array		contains a list of IP addresses.

*** Indicates a field that is omitted when you enable privacy controls.**

Application and File threat fields

These fields exist in the details of Lookout threat events where event.details.type=APPLICATION or =FILE.

Lookout event field	Type	ArcSight (CEF) field	Description
event.details.applicationName	String	deviceProcessName	Name of the application.
event.details.packageName	String	deviceProcessName	Package name of the application.
event.details.fileName	String	fname	Name of the file.
event.details.path	String	filePath	Path of the file.

Configuration threat fields

These fields exist in the details of Lookout threat events where event.details.type=CONFIGURATION.

Lookout event field	Type	ArcSight (CEF) field	Description
event.details.TrustedSigningIdentity	String	destinationServiceName	The detected third party signing identity on the device, ex: “iPhone Distribution: Known Malware Co., Ltd”

OS threat fields

These fields exist in the details of Lookout threat events where event.details.type=OS.

Lookout event field	Type	ArcSight (CEF) field	Description
event.details.osVersion	String	flexString1	Custom field osVersion with the version of the OS, e.g., “10.3”.

Device event common fields

These fields exist in the details of Lookout events where event.type=DEVICE.

Lookout event field	Type	ArcSight (CEF) field	Description
event.details.	Enum	cs5	Custom field activationStatus with a value from

activationStatus			[ACTIVATED, DEACTIVATED, PENDING, DELETED] indicating device activation status .
event.details. protectionStatus	Enum	cs3	Custom field protectionStatus with a value from [PROTECTED, DISCONNECTED] indicating device protection status.
event.details. securityStatus	Enum	cs4	Custom field securityStatus with a value from [SECURE, THREAT_LOW, THREAT_MEDIUM, THREAT_HIGH] indicating device security status
event.details. updatedDetails	Array	reason	Indicates the device status changes triggering the event. Can be any combination of [activationStatus, protectionStatus, securityStatus]

Device target fields

These fields exist in the target details of individual Lookout device events.

Lookout event field	Type	ArcSight (CEF) field	Description
event.target.id	String	suid	Lookout's unique, internal identifier for the device.
event.target.externalId	String	cs6	Custom field externalId with a unique, application-specific external ID. It can correspond to your MDM or MAM device ID.
event.target.platform	Enum	sourceServiceName	Indicates the target platform as [Android, iOS, Other].
*event.target.emailAddress	String	suser	Email address of device owner.

*** Indicates a field that is omitted when you enable privacy controls.**

QRadar (LEEF) fields

This section shows the mapping of Lookout event fields into QRadar (LEEF) fields.

Common fields

These fields are present in all Lookout events.

Lookout event field	Type	QRadar (LEEF) field	Description
n/a	String	Vendor	Static value of "Lookout" supplied.

n/a	String	`Product`	Static value of “SIEM Client” supplied.
n/a	String	`Version`	*<Lookout SIEM client version>*
n/a	String	`lookoutEntName`	*<Entity name for event identification>*
`event.type`, `event.details.type`	EventType, EventDetailsType	`EventID`	The event ID derived from a concatenation of the Lookout event.type (`THREAT` or `DEVICE`) and event.details.type (see Threat event common fields below).
`event.id`	String	`lookoutId`	Unique ID of the event.
`event.eventTime`	DateTime	`lookoutEventTime`	Date and time of the event.

Threat event common fields

These fields exist in the details of all Lookout events where `event.type=THREAT`.

Lookout event field	Type	QRadar (LEEF) field	Description
`event.details.type`	Enum	`cat`	Indicates a threat event type of `[APPLICATION,NETWORK, FILE,OS]`.
`event.details.id`	GUID	`lookoutThreatId`	Lookout’s internal GUID for each threat.
`event.details.action`	Enum	`lookoutAction`	Indicates the state of the threat action as `[DETECTED, RESOLVED, IGNORED]`.
`event.details.severity`	Enum	`lookoutSeverity`	Indicates the severity of the threat event as `[LOW, MEDIUM, HIGH]`.
`event.details.classifications`	Array	`lookoutClassifications`	Indicates the classification type of the threat event. Can be any of `[ROOT_ENABLER, RISKWARE, ADWARE, CHARGEWARE, DATA_LEAK, TROJAN, WORM, VIRUS, EXPLOIT, BACK_DOOR, BOT, APP_DROPPER, CLICK_FRAUD, SPAM, SPYWARE, SURVEILLANCEWARE, VULNERABILITY, ROOT_JAIL_BREAK, CONNECTIVITY, TOLL_FRAUD, SIDELOADED_APP, ACTIVE_MITM, UNKNOWN]`.

Network threat fields

These fields exist in the details of Lookout threat events where `event.details.type=NETWORK`.

Lookout event field	Type	QRadar (LEEF) field	Description
***event.details.ssid**	String	`lookoutSSID`	The wireless network ID.
***event.details.macAddress**	String	`lookoutMacAddress`	The MAC address.
`event.details.proxyProtocol`	String	`lookoutProxyProtocol`	The proxy protocol being used.
`event.details.proxyPort`	Integer	`srcPostNatPort`	The proxy port being used.
`event.details.proxyAddress`	String	`lookoutProxyAddress`	The proxy IP address.
`event.details.vpnLocalAddress`	String	`lookoutVpnLocalAddress`	The VPN IP address.
`event.details.vpnPresent`	Boolean	`lookoutVpnPresent`	Indicates if a VPN exists.
`event.details.dnsIpAddresses`	String array	`lookoutDnsIpAddress`	List of IP addresses.

*** Indicates a field that is omitted when you enable privacy controls.**

Application and File threat fields

These fields exist in the details of Lookout threat events where `event.details.type=APPLICATION` or `=FILE`.

Lookout event field	Type	QRadar (LEEF) field	Description
`event.details.applicationName`	String	`lookoutAppProcessDetails`	Name of the application.
`event.details.packageName`	String	`lookoutAppProcessDetails`	Package name of the application.
`event.details.fileName`	String	`lookoutAppFileName`	Name of the file.
`event.details.path`	String	`lookoutAppFilePath`	Path of the file.

Configuration threat fields

These fields exist in the details of Lookout threat events where `event.details.type=CONFIGURATION`.

Lookout event field	Type	QRadar (LEEF) field	Description
event.details. TrustedSigningIdentity	String	lookoutTrustedSigningIdentity	The detected third party signing identity on the device, ex: “iPhone Distribution: Known Malware Co., Ltd”

OS threat fields

These fields exist in the details of Lookout threat events where `event.details.type=OS`.

Lookout event field	Type	QRadar (LEEF) field	Description
event.details. osVersion	String	lookoutOSVersion	Version of the OS, e.g., “10.3”.

Device event common fields

These fields exist in the details of Lookout device events where `event.type=DEVICE`.

Lookout event field	Type	QRadar (LEEF) field	Description
event.details. activationStatus	Enum	lookoutActivationStatus	Indicates a device activation status of [ACTIVATED, DEACTIVATED, PENDING, DELETED].
event.details. protectionStatus	Enum	lookoutProtectionStatus	Indicates a device protection status of [PROTECTED, DISCONNECTED].
event.details. securityStatus	Enum	lookoutSecurityStatus	Indicates a device security status of [SECURE, THREAT_LOW, THREAT_MEDIUM, THREAT_HIGH].
event.details. updatedDetails	Array	lookoutUpdatedDetails	Indicates the device status changes triggering the event. Can be any combination of [activationStatus, protectionStatus, securityStatus]

Device target fields

These fields exist in the target details of individual Lookout device events.

Lookout event field	Type	QRadar (LEEF) field	Description
event.target. id	String	lookoutTargetId	Lookout’s unique, internal identifier for the device.
event.target.	String	lookoutTargetExternalId	A unique, application-specific

externalId			external ID. It can correspond to your MDM or MAM device ID.
event.target.platform	Enum	lookoutTargetPlatform	Indicates the target platform as [Android, iOS, Other].
*event.target.emailAddress	String	lookoutUser	Email address of device owner.

*** Indicates a field that is omitted when you enable privacy controls.**

Splunk (KEY-VALUE) fields

This section shows the key-value mapping of Lookout event fields into Splunk fields.

Common fields

These fields are present in all Lookout events.

Lookout event field	Type	Splunk field	Description
event.type	Event type	type	Indicates the event type as [THREAT, DEVICE].
event.id	String	id	Unique ID identifying the event.
event.eventTime	DateTime	eventTime	Date and time of the event.
n/a	String	entName	*<Entity name for events identification>*.

Threat event common fields

These fields exist in the details of Lookout events where event.type=THREAT.

Lookout event field	Type	Splunk field	Description
event.details.type	Enum	details.type	Indicates a threat event type of [APPLICATION,NETWORK, FILE,OS].
event.details.id	GUID	details.id	Lookout's internal GUID for each threat.
event.details.action	Enum	details.action	Indicates the state of the threat action as [DETECTED, RESOLVED, IGNORED].
event.details.severity	Enum	details.severity	Indicates the severity of the threat event as [LOW, MEDIUM, HIGH].

event.details.classifications	Array	details.classifications	Indicates the classification type of the threat event. Can be any of [ROOT_ENABLER, RISKWARE, ADWARE, CHARGEWARE, DATA_LEAK, TROJAN, WORM, VIRUS, EXPLOIT, BACK_DOOR, BOT, APP_DROPPER, CLICK_FRAUD, SPAM, SPYWARE, SURVEILLANCEWARE, VULNERABILITY, ROOT_JAIL_BREAK, CONNECTIVITY, TOLL_FRAUD, SIDELOADED_APP, ACTIVE_MITM, UNKNOWN].

Network threat fields

These fields exist in the details of Lookout threat events where event.details.type=NETWORK.

Lookout event field	Type	Splunk field	Description
*event.details.ssid	String	details.ssid	The wireless network ID.
*event.details.macAddress	String	details.macAddress	The MAC address.
event.details.proxyProtocol	String	details.proxyProtocol	The proxy protocol being used.
event.details.proxyPort	Integer	details.proxyPort	The proxy port being used.
event.details.proxyAddress	String	details.proxyAddress	The proxy IP address.
event.details.vpnLocalAddress	String	details.vpnLocalAddress	The VPN IP address.
event.details.vpnPresent	Boolean	details.vpnPresent	Indicates if a VPN exists.
event.details.dnsIpAddresses	String array	details.dnsIpAddresses	List of IP addresses.

*** Indicates a field that is omitted when you enable privacy controls.**

Application and File threat fields

These fields exist in the details of Lookout threat events where event.details.type=APPLICATION or =FILE.

Lookout event field	Type	Splunk field	Description
event.details.applicationName	String	details.applicationName	Name of the application.

event.details.packageName	String	details.packageName	Package name of the application.
event.details.fileName	String	details.fileName	Name of the file.
event.details.path	String	details.path	Path of the file.

Configuration threat fields

These fields exist in the details of Lookout threat events where event.details.type=CONFIGURATION.

Lookout event field	Type	Splunk field	Description
event.details.TrustedSigningIdentity	String	details.trustedSigningIdentity	The detected third party signing identity on the device, ex: “iPhone Distribution: Known Malware Co., Ltd”

OS threat fields

These fields exist in the details of Lookout threat events where event.details.type=OS.

Lookout event field	Type	Splunk field	Description
event.details.osVersion	String	details.osVersion	Version of the OS, e.g., “10.3”.

Device event common fields

These fields exist in the details of Lookout device events where event.type=DEVICE.

Lookout event field	Type	Splunk field	Description
event.details.activationStatus	Enum	details.activationStatus	Indicates a device activation status of [ACTIVATED, DEACTIVATED, PENDING, DELETED].
event.details.protectionStatus	Enum	details.ProtectionStatus	Indicates a device protection status of [PROTECTED, DISCONNECTED].
event.details.securityStatus	Enum	details.SecurityStatus	Indicates a device security status of [SECURE, THREAT_LOW, THREAT_MEDIUM, THREAT_HIGH].
event.details.updatedDetails	Array	details.UpdatedDetails	Indicates the device status changes triggering the event. Can be any combination of [activationStatus, protectionStatus, securityStatus]

Device target fields

These fields exist in the target details of individual Lookout device events.

Lookout event field	Type	Splunk field	Description
`event.target.type`	Enum	`target.type`	Indicates the target type of `DEVICE`.
`event.target.id`	String	`target.id`	Lookout's unique, internal identifier for the device.
`event.target.externalId`	String	`target.externalId`	A unique, application-specific external ID. It can correspond to your MDM or MAM device ID.
`event.target.platform`	Enum	`target.platform`	Indicates the target platform as `[Android, iOS, Other]`.
***event.target.emailAddress**	String	`target.emailAddress`	Email address of device owner.
`event.target.links`	Link object	`target.links`	Link to retrieve the device object. Valid when a device endpoint is available.

*** Indicates a field that is omitted when you enable privacy controls.**

Errors

The SIEM connector writes internal errors to the log file specified in the `app.properties` configuration file (see the Configuration files section). Errors are also written to your SIEM instance, unless the error relates to the connector not contacting your SIEM instance.

ArcSight (CEF) Error Mappings

ArcSight (CEF) field	Type	Description
`deviceVendor` (header)	String	Static value of “Lookout” written.
`deviceProduct` (header)	String	Static value of “SIEM Client” written.
`deviceVersion` (header)	String	*<Lookout SIEM client version>*
`deviceEventClassID` (header)	String	Static value of “CLIENT” written.
`name` (header)	Event type	Static value of “ERROR” written.
`severity` (header)	String	Static value of “HIGH” written.
`start`	DateTime	Date and time of the event.
`devicePayloadId`	String	Unique ID identifying the error event.
`cs1`	String	Entity name for the error event.
`msg`	String	Description of the error.

The following is an example of an ArcSight error event.

```
CEF:0|Lookout|SIEM Client|0.1|CLIENT|ERROR|HIGH|
devicePayloadId=437439f8-fb4a-43c0-95c5-4cc2a9688f27 start=2017-07-22T12:45:07
cs1=testEntName msg=Error while polling events from Lookout Mobile Risk API
```

QRadar (LEEF) Error Mappings

QRadar (LEEF) field	Type	Description
`vendor` (header)	String	“Static value of “Lookout” written.
`product` (header)	String	Static value of “SIEM Client” written.
`version` (header)	String	*<Lookout SIEM client version>*

`eventId` (header)	Event type	Static value of “ERROR,CLIENT” written.
`lookoutEventTime`	DateTime	Date and time of the event.
`lookoutId`	String	Unique ID identifying the error event.
`lookoutEntName`	String	Entity name for the error event.
`lookoutMsg`	String	Description of the error.
`cat`	String	Static value of “CLIENT” written.
`lookoutSeverity`	String	Static value of “HIGH” written.

The following is an example of a QRadar error event.

```
LEEF:1.0|Lookout|SIEM Client|0.1|ERROR,CLIENT|lookoutSeverity=HIGH\t
lookoutId=437439f8-fb4a-43c0-95c5-4cc2a9688f27\tcat=CLIENT\t
lookoutEventTime=2017-03-22T12:45:07\tlookoutMsg=Error while polling
events from Lookout Mobile Risk API \tlookoutEntName=testEntName
```

Splunk (KEY_VALUE) Error Mappings

Splunk field	Type	Description
`type`	Event type	Static value of “ERROR” written.
`id`	String	Unique ID identifying the error event.
`eventTime`	DateTime	Date and time of the error event.
`entName`	String	Entity name for the error event.
`msg`	String	Description of the error.

The following is an example of a Splunk error event.

```
type=ERROR, id=437439f8-fb4a-43c0-95c5-4cc2a9688f27,
eventTime=2017-03-22T12:45:07, entName=testEntName, msg=Error while
polling events from Lookout Mobile Risk API
```