

In-Sight Platform API

API Version	Description
1.0	Released with In-Sight 24.3.0
1.1	Released with In-Sight 25.1.0: Add support for custom server certificates
1.2	Released with In-Sight 25.2.0: Add support for syslog forwarding TCP framing option

Document Revision 25.2.0.64

Table of Contents

- 1. Overview..... 3
- 2. Platform API..... 3
 - 2.1 Root Resources ("api")..... 3
 - 2.1.1 Audit Log..... 3
 - 2.1.2 Server Certificates..... 9
- 3. Object Types..... 10
 - 3.1 Audit Log..... 10
 - 3.1.1 Priority..... 11
 - 3.1.2 Structured Data..... 11
 - 3.1.2.1 Source..... 11
 - 3.1.2.2 Category..... 12
 - 3.1.3 Syslog Forwarding..... 12
 - 3.2 CertificateMetadataListModel..... 12
 - 3.2.1 CertificateMetadataModel..... 12
 - 3.2.2 X509NameModel..... 13
- 4. Error Handling..... 13
 - 4.1 HTTP Responses..... 13

1. Overview

The In-Sight Platform API uses the HTTP protocol to access and change resource values on the camera. This document describes the Platform API that is available via that protocol.

2. Platform API

In the following section, HTTP requests are made to resources.

2.1 Root Resources (“api”)

The following items’ base resource is “api” (e.g. “api/audit-log”).

Example usage with curl:

```
curl --user admin: --request GET http://127.0.0.1:80/api/audit-log
```

Default username is “admin” and password “”. The user must have an access level of “Full” to be authorized to make requests to these resources.

2.1.1 Audit Log

Endpoint	Description		
audit-log	GET – Get the audit log.		
	Example usage with curl: curl --user admin: --request GET http://127.0.0.1:80/api/audit-log? before=Value&after=Value --header "Accept-Encoding: gzip"		
	Query Parameters		
	Parameter	Value	Default
	before	ISO 8601 formatted date	N/A
	after	ISO 8601 formatted date	N/A
	Only events with a timestamp after the provided date are included in the response. When omitted, there is no "after" filtering applied.		
	Supported Coding Types (Compression)		
	Request “Accept-Encoding” Header	Response “Content-Encoding” Header	Description

	"Accept-Encoding:" (or not given)	N/A	No encoding requested; response will yield uncompressed JSON data.
	"Accept-Encoding: gzip"	"Content-Encoding: gzip"	"gzip" encoding requested; response will yield compressed JSON data, which can be decompressed with "gzip -d" or similar.

Response Data

JSON array literal of [Audit Log](#) objects.

```

{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "Download audit log, GET /api/audit-log",
  "type": "array",
  "items": {
    "description": "Download audit log event entry",
    "type": "object",
    "required": [
      "timestamp",
      "hostname",
      "severity",
      "type",
      "event",
      "result",
      "structuredData"
    ],
    "properties": {
      "timestamp": {
        "description": "ISO 8601 formatted timestamp of the event",
        "type": "string",
        "format": "date-time"
      },
      "hostname": {
        "description": "Hostname of the machine the event occurred on",
        "type": "string",
        "minLength": 1
      },
      "severity": {
        "description": "Log severity of the event",
        "type": "string",
        "enum": [
          "emerg",
          "alert",
          "crit",
          "err",
          "warning",
          "notice",
          "info",
          "debug"
        ]
      },
      "type": {
        "description": "Audit log event type",
        "type": "string",
        "minLength": 1
      },
      "event": {
        "description": "Audit log event id",
        "type": "string",
        "minLength": 1
      },
      "result": {
        "description": "Audit log event result",
        "type": "string",
        "minLength": 0
      }
    }
  }
}

```

```

"structuredData": {
  "description": "Structured data in the corresponding syslog messages",
  "required": [
    "auditEvent@39975",
    "timeQuality"
  ],
  "properties": {
    "auditEvent@39975": {
      "description": "Additional IEC 62443 4-2 audit event data",
      "required": [
        "category",
        "source",
        "id"
      ],
      "additionalProperties": false,
      "properties": {
        "source": {
          "description": "Actor which triggered the audit log event",
          "type": "string",
          "enum": [
            "device",
            "remoteSystem",
            "humanUser"
          ]
        },
        "category": {
          "description": "Audit log event category",
          "type": "string",
          "enum": [
            "accessControl",
            "requestError",
            "controlSystem",
            "backupRestore",
            "config",
            "auditLog"
          ]
        },
        "id": {
          "description": "Id of the audit event resetting to 0 each boot and incrementing
by 1 for each successive event",
          "type": "integer",
          "minimum": 0,
        }
      }
    },
    "timeQuality": {
      "description": "RFC 5424 timeQuality structured data parameters",
      "required": [
        "isSynced",
        "tzKnown"
      ],
      "properties": {
        "isSynced": {
          "description": "Identifies whether the timestamp was recorded while the device
was synchronized with a time server",
          "type": "string",
          "enum": [
            "0",
            "1"
          ]
        },
        "tzKnown": {
          "description": "Identifies whether the timestamp was set with the time zone
configured",
          "type": "string",
          "enum": [
            "0",
            "1"
          ]
        }
      }
    },
    "auth@39975": {
      "description": "Describe the authorization of an event - who did it",
      "required": [
        "user"
      ]
    }
  ]
}

```

	<pre> "additionalProperties": false, "properties": { "user": { "description": "Id of coguser which was authorized for the event", "type": "string", "minLength": 1 } }, "configChange@39975": { "description": "Describe the configuration change", "required": ["old", "new"], "additionalProperties": false, "properties": { "property": { "description": "Description of the property being changed if the 'event id' is too constrained to adequately identify the system property being configured.", "type": ["string", "number", "integer", "object", "array", "boolean", "null"] }, "old": { "description": "Value used for the configuration option prior to the config change event", "type": ["string", "number", "integer", "object", "array", "boolean", "null"] }, "new": { "description": "New value used for the configuration option as a result of the config change event", "type": ["string", "number", "integer", "object", "array", "boolean", "null"] } } } </pre>
audit-log/ syslog- forwarding	GET – Get the current syslog forwarding configuration of whether the forwarding is enabled. When enabled, also include the address, port, and severity filter. Example usage with curl: curl --user admin: --request GET http://127.0.0.1:80/api/audit-log/syslog-forwarding Response Data

```

{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "GET /api/audit-log/syslog-forwarding response data",
  "type": "object",
  "required": [
    "enabled"
  ],
  "additionalProperties": false,
  "properties": {
    "enabled": {
      "type": "boolean"
    },
    "address": {
      "type": "string",
      "anyOf": [
        {
          "format": "idn-hostname"
        },
        {
          "format": "ipv4"
        }
      ],
      "description": "Address of the remote syslog collector which audit events will be forwarded to"
    },
    "port": {
      "type": "integer",
      "minimum": 1,
      "maximum": 65535
    },
    "severityFilter": {
      "description": "Log severity of the event",
      "type": "string",
      "enum": [
        "emerg",
        "alert",
        "crit",
        "err",
        "warning",
        "notice",
        "info",
        "debug"
      ]
    },
    "tcp_framing": {
      "description": "TCP framing mode",
      "type": "string",
      "enum": [
        "octet-counted",
        "traditional"
      ],
      "default": "octet-counted"
    }
  ],
  "allof": [
    {
      "if": {
        "properties": {
          "enabled": {
            "const": true
          }
        }
      },
      "then": {
        "required": [
          "severityFilter",
          "port",
          "address",
          "tcp_framing"
        ]
      }
    }
  ],
  "examples": [
    {
      "enabled": false
    }
  ],

```

```

    {
      "enabled": true,
      "address": "example-server-hostname",
      "port": 6514,
      "severityFilter": "warning",
      "tcp_framing": "octet-counted"
    }
  ]
}

```

PUT – Enable syslog forwarding by specifying the remote forwarding address and optionally an event severity filter. *Note: Syslog forwarding requires octet-counting TCP framing in the syslog collector.*

Example usage with curl:

```
curl --user admin: --request PUT --header "Content-Type: application/json"
--data "Request Data" http://127.0.0.1:80/api/audit-log/syslog-forwarding
```

Request Data

```

{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "PUT /api/audit-log/syslog-forwarding request data",
  "type": "object",
  "required": [
    "address"
  ],
  "additionalProperties": false,
  "properties": {
    "severityFilter": {
      "description": "Log severity of the event",
      "type": "string",
      "enum": [
        "emerg",
        "alert",
        "crit",
        "err",
        "warning",
        "notice",
        "info",
        "debug"
      ],
      "default": "info"
    },
    "port": {
      "type": "integer",
      "minimum": 1,
      "maximum": 65535,
      "default": 6514
    },
    "address": {
      "type": "string",
      "anyOf": [
        {
          "format": "idn-hostname"
        },
        {
          "format": "ipv4"
        }
      ],
      "description": "Address of the remote syslog collector which audit events will be forwarded to"
    },
    "tcp_framing": {
      "description": "TCP framing mode",
      "type": "string",
      "enum": [
        "octet-counted",
        "traditional"
      ],
      "default": "octet-counted"
    }
  },
  "examples": [
    {
      "address": "example-server-hostname",

```

	<pre> "port": 6514, "severityFilter": "info", "tcp_framing": "octet-counted" }] } </pre>
	DELETE – Disable syslog forwarding. Example usage with curl: curl --user admin: --request DELETE http://127.0.0.1:80/api/audit-log/syslog-forwarding

2.1.2 Server Certificates

Note: The Server Certificates API is currently under active development. **Breaking changes** are expected in the coming releases as we continue to enhance and refine its functionality.

The following endpoints allow users to create, read, and delete server certificates on the device. A certificate signing request must be submitted before installing a new certificate. There is no way to upload separate public and private keys; private keys do not leave the device.

If INSIGHT_TLS_CERT is used for the certificate ID (denoted as {id} in the table below), then the certificate will be used for all HTTPS traffic. When the certificate is installed, the device automatically picks up the new certificate and restarts the web server. When the certificate is deleted, the device automatically falls back to the self-signed certificate and restarts the web server.

Endpoint	Verb Description
security/certificates/tls	GET – Get the list of installed certificates. Example usage with curl: curl --user admin: --request GET http://127.0.0.1:80/api/security/certificates/tls Response Data 200 – JSON object literal of CertificateMetadataListModel
security/certificates/tls/{id}	GET – Get the certificate. Example usage with curl: curl --user admin: --request GET http://127.0.0.1:80/api/security/certificates/tls/INSIGHT_TLS_CERT Response Data 200 – JSON object literal of CertificateMetadataModel 404 – TLS certificate not found
	DELETE – Delete the certificate. Example usage with curl: curl --user admin: --request DELETE

	<code>http://127.0.0.1:80/api/security/certificates/tls/INSIGHT_TLS_CERT</code> Response Data 204 – TLS certificate deleted 400 – Invalid TLS certificate identity 404 – TLS certificate not found
<code>security/certificates/tls/{id}/csr</code>	PUT – Generate a certificate signing request for the device using a 3072-bit RSA key. Example usage with curl: <pre>curl --user admin: --request PUT --header "Content-Type: application/json" --data '{"subject": {"C": "US", "ST": "MA", "L": "Natick", "O": "Cognex", "CN": "IS3805MP-abcdef"}, "subjectAltName": ["DNS:IS3805MP-abcdef.net.corporation.com", "IP:127.0.0.1"]}' http://127.0.0.1/api/security/certificates/tls/INSIGHT_TLS_CERT/csr</pre> Response Data 200 - String containing the contents of a certificate signing request 400 – Invalid CSR identity
<code>security/certificates/tls/{id}/pem</code>	PUT – Install a signed certificate. Example usage with curl: <pre>curl -user admin: --request PUT --form cert_pem_file=@/data/workspace/certificate-management/signed.pem http://127.0.0.1/api/security/certificates/tls/INSIGHT_TLS_CERT/pem</pre> Response Data 200 – JSON object literal of CertificateMetadataModel 400 – Invalid request

3. Object Types

3.1 Audit Log

Property	Description
severity	Priority
type	Customer facing ID for the subsystem or process on the system that records the event.
event	The event ID uniquely identifies an audit log event within the scope of the audit log event type.
result	When an event ID may not sufficiently describe the result of an event due to non-static data, additional data may be provided in the event result. Events are allowed to have an empty event result.
structuredData	Structured Data
hostname	Name of the device on which the event occurred.
timestamp	System time in milliseconds when the log event was reported.

3.1.1 Priority

Level	Value	Example Usage
Emergency	emerg	N/A
Alert	alert	- Detected tamper events - Detected configuration incompatible with security policy
Critical	crit	N/A
Error	err	Failure to apply configuration changes
Warning	warning	Access denied to device
Notice	notice	Changes in system configuration
Informational	info	Access granted to device
Debug	debug	N/A

3.1.2 Structured Data

SD-ID	PARAM-NAME	PARAM-VALUE
timeQuality	tzKnown	See RFC-5424: The Syslog Protocol, section 7.1.1 tzKnown.
	IsSynced	See RFC-5424: The Syslog Protocol, section 7.1.2 isSynced.
auditEvent@39975	source	Source
	category	Category
	id	Monotonically increasing counter to track all successfully recorded audit events.
auth@39975	user	ID of coguser that was authorized for the event.
configChange@39975	property	Description of the property being changed if the event ID is too constrained to adequately identify the system property being configured.
	old	Value used for the configuration option prior to the config change event.

	new	New value used for the configuration option because of the config change event.
--	-----	---

3.1.2.1 Source

Name	Value	Actor triggering the event
Originating Device	device	A process internal to the system
Software Process	remoteSystem	A remote system without human interaction
Human User Account	humanUser	A human interaction with the system

3.1.2.2 Category

Name	Value	Priority	Event Trigger
Access Control	accessControl	1	An access request to the device by any external person or system has been granted or denied. - Password/certificate-based authentication success/failure to the device - Access to the device granted/denied due to user permissions
Request Error	requestError	6	An error occurred while processing a request from a remote actor. Invalid request data
Control System Event	controlSystem	4	The device sends a request to a remote system to alter the behavior of that remote system.
Backup and Restore Event	backupRestore	5	- A backup download started - A backup download failed - A restore started - A restore completed - A restore failed
Configuration Change	config	3	The configuration of the system has been changed. - Network interface IP address changes - Common service configuration changes
Audit Log	auditLog	2	Any event that impacts the quantity, quality, or availability of information available in the audit log.

3.1.3 Syslog Forwarding

Property	Type	Description
enabled	boolean	Whether or not syslog forwarding is enabled. If enabled is false, then no other properties are returned in the GET request.
address	string	The address of the server to forward logs to.
port	integer	The port of the server to forward logs to.
severityFilter	string	The minimum severity threshold for log forwarding, where messages matching or exceeding this severity level are forwarded while less severe messages are excluded.
tcp_framing	string	The TCP framing mode to use, accepting either "octet-counted" (default) or "traditional" framing methods to determine message boundaries.

3.2 CertificateMetadataListModel

Property	Type	Description
{id}	CertificateMetadataModel	Certificate information.

3.2.1 CertificateMetadataModel

Property	Type	Description
serialNumber	integer	Big-endian hexadecimal string set by the Certificate Authority.
version	integer	X509 version of the certificate.
subject	X509NameModel	Distinguished name of the client to which the certificate belongs.
issuer	X509NameModel	Distinguished name of the issuing Certificate Authority that signed the certificate.
notBefore	date-time (string)	ASN1_TIME of when the certificate becomes valid.
notAfter	date-time (string)	ASN1_TIME of when the certificate expires.
subjectAltName	array<string>	List of additional host names (IP addresses, common names, etc.) to be protected by this certificate. Values must be prefixed with IP: or DNS: depending on the type.
subjectHash	string	Hash value of the subject of the certificate.
expired	boolean	Whether or not the certificate has expired.

3.2.2 X509NameModel

Property	Type	Description
C	string	Two-letter ISO country code (e.g., US, CA)
ST	string	Geographic region within country
L	string	City or town name
O	string	Legal entity name (e.g., Example Corporation)
OU	string	Department or division (e.g., IT Department)
CN	string	Device name or device IP

4. Error Handling

When there is an error completing the request, an error response is returned:

```
{
  "status": <http error response code>,
  "detail": "Example description of a user displayed error message for the bad request"
}
```

4.1 HTTP Responses

If you are making HTTP requests for the resources, then the expected response would look like this...

A successful GET:
curl --user admin: http://127.0.0.1:80/api/audit-log

200 OK Response with JSON payload in the body

When a page not found:

```
curl --user admin: http://127.0.0.1:80/api/audit-log2
```

HTTP ERROR 404 Not Found

```
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

Access to a resource with unauthorized user:

```
curl --user John: http://127.0.0.1:80/api/audit-log
```

HTTP ERROR 401 Unauthorized

```
<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx</center>
</body>
</html>
```