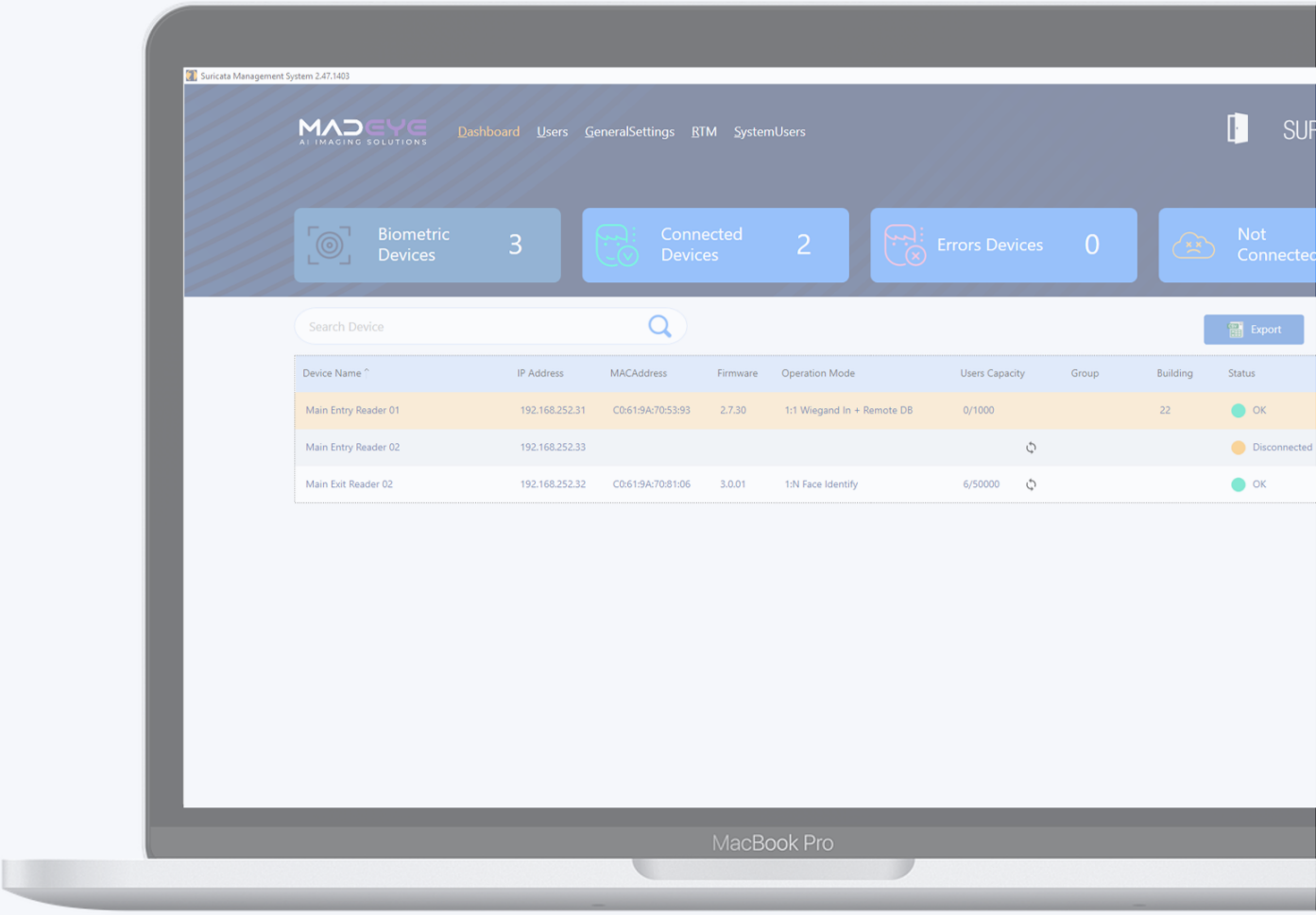
SURICATA

User Guide



Version: 1.1
Date: May 2024

MADEYE
AI IMAGING SOLUTIONS

Contents

Overview **3**
Architecture 3
Installation **4**
Prerequisites 4
Suricata Server Installation Guide 6
Suricata Client Installation Guide 8
Dashboard **13**
Add a New Device 14
Edit or Delete a Device 15
Configure Device Settings 16
Users **23**
Add a New User 24
Encode a User's Badge 26
Edit or Delete a User 26
General Settings **27**
Add a New Profile/Setting 27
Edit or Delete a Profile/Setting 27
Real-Time Monitoring (RTM) **29**
System Users **30**
Add a New System User 30
Edit or Delete a System User 31
System Roles 32
API **33**
General 33
Connection Details 33
API Functions 33
API Events for Callback Registration 34
Schema Classes 35
Code Samples 36
Logs 37

Overview

Suricata is a software management tool for Madeye VisionA-64™ face recognition terminals.

The Suricata software package includes two applications:

1. Suricata Server: This application runs in the background and is required to connect the client to the face recognition terminals.
2. Suricata Client: This is the application users will access to monitor the system, configure settings, and manage badges.

Architecture

Suricata Server: Windows Service and database

Suricata Client: Windows GUI

Multiple clients can be connected to the Suricata Server.

The Suricata package includes an API for the server component allowing easy and simple integration for user management and RTM events registration.

Installation

> **Note:**
>
> To run Suricata, you will need to install two applications:
>
> 1. Suricata Server
> 2. Suricata Client

Prerequisites

Suricata Server Hardware Requirements

The following memory and processor requirements apply to all editions of Suricata Server:

Component	Requirement
Storage	Minimum: 16 GB of available hard drive space
Monitor	Super-VGA (800x600) or higher resolution
Memory	Minimum: 1 GB Recommended: 4 GB
Processor Speed	Minimum: 1.4 GHz x64 Recommended: 2.0 GHz or faster
Processor Type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support

Suricata Server Software Requirements

The following software requirements apply to all Suricata Server installations:

Component	Requirement
Operating System	Windows 10 1607 or later Windows Server 2016 or later
.NET Framework	Required

Suricata Client Hardware Requirements

The following memory and processor requirements apply to all editions of Suricata Client:

Component	Requirement
Storage	Minimum: 16 GB of available hard drive space
Monitor	Super-VGA (1920x1200) or higher resolution
Memory	Minimum: 1 GB Recommended: 4 GB
Processor Speed	Minimum: 1.4 GHz x64 Recommended: 2.0 GHz or faster
Processor Type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support

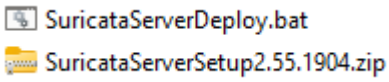
Suricata Server Software Requirements

The following software requirements apply to all Suricata Client installations:

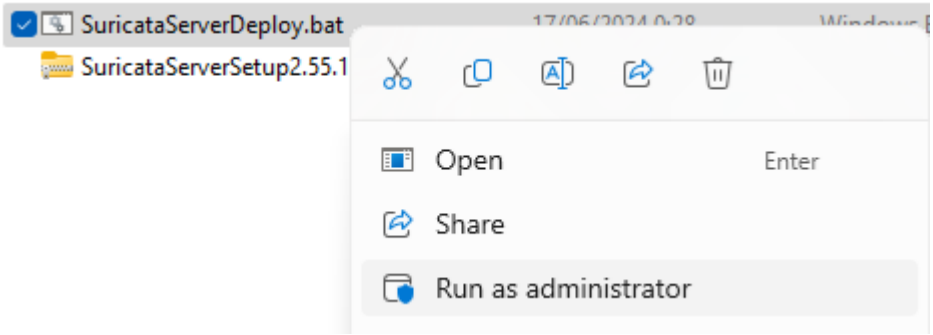
Component	Requirement
Operating System	Windows 10 1607 or later Windows Server 2016 or later
.NET Framework	Required

Suricata Server Installation Guide

1. Download the installation files (SuricataServerSetup.zip and SuricataServerDeploy.bat) to the computer you want to install Suricata Server on.

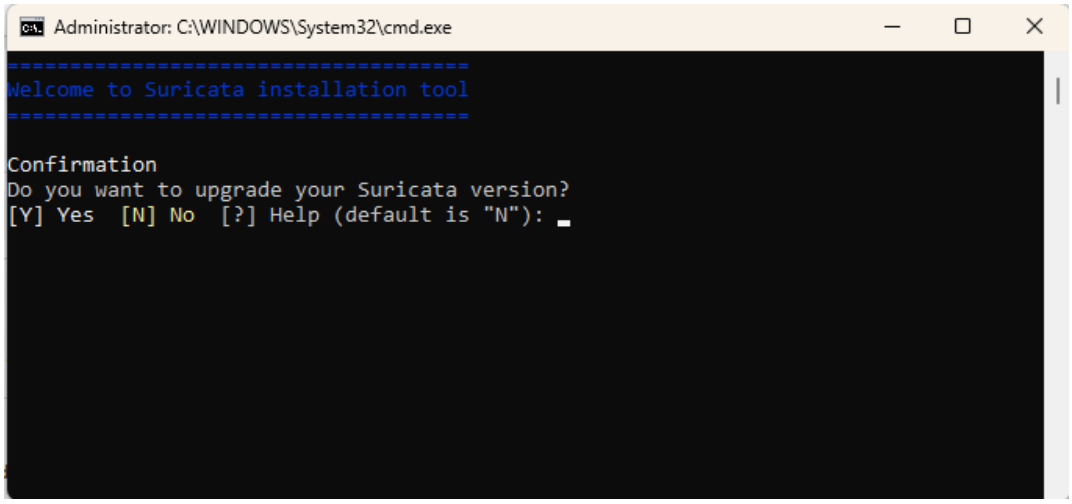


2. Right click on SuricataServerDeploy.bat and select **Run as administrator**.

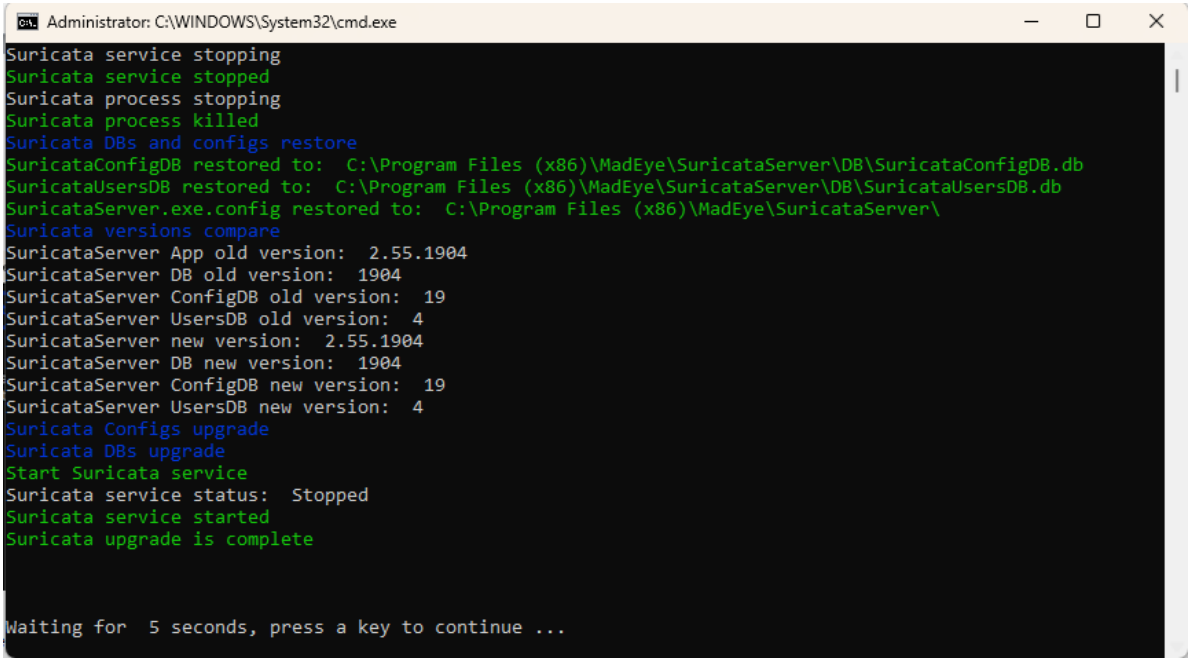


3. The notify message will appear. Click **Yes**.

4. The confirmation message will appear. Click **Y** and then click **Enter.**

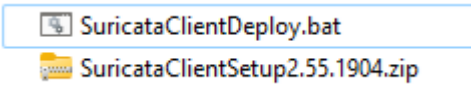


5. Once the installation is complete, the command window will close itself after 10 seconds

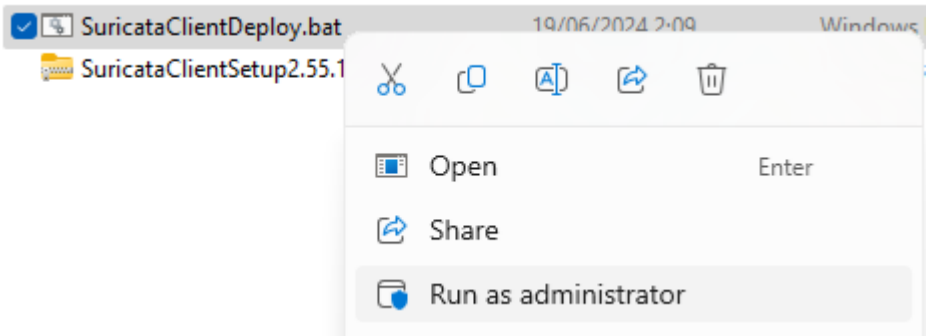


Suricata Client Installation Guide

1. Download the installation files (SuricataClientSetup.zip and SuricataClientDeploy.bat) to the computer you want to install Suricata Client on.

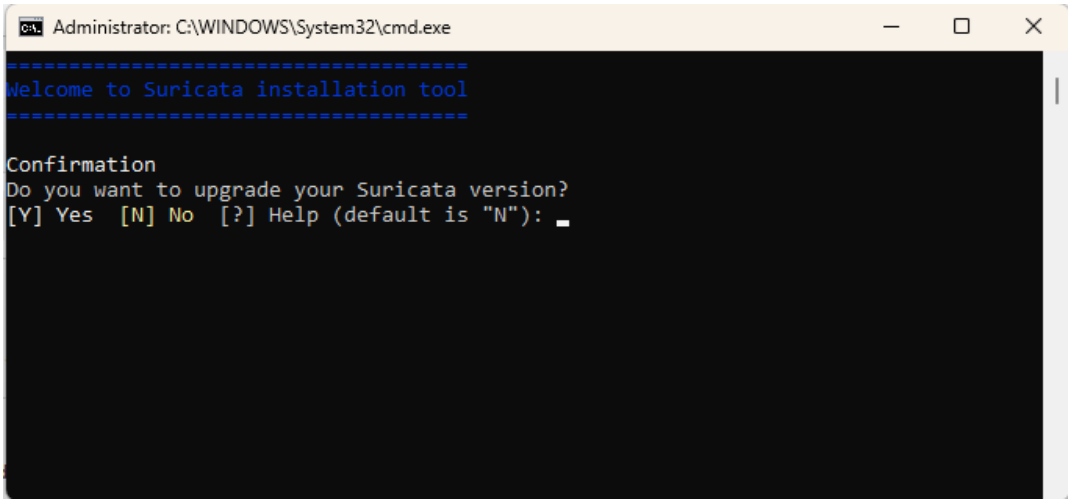


2. Right click on SuricataClientDeploy.bat and select **Run as administrator**.

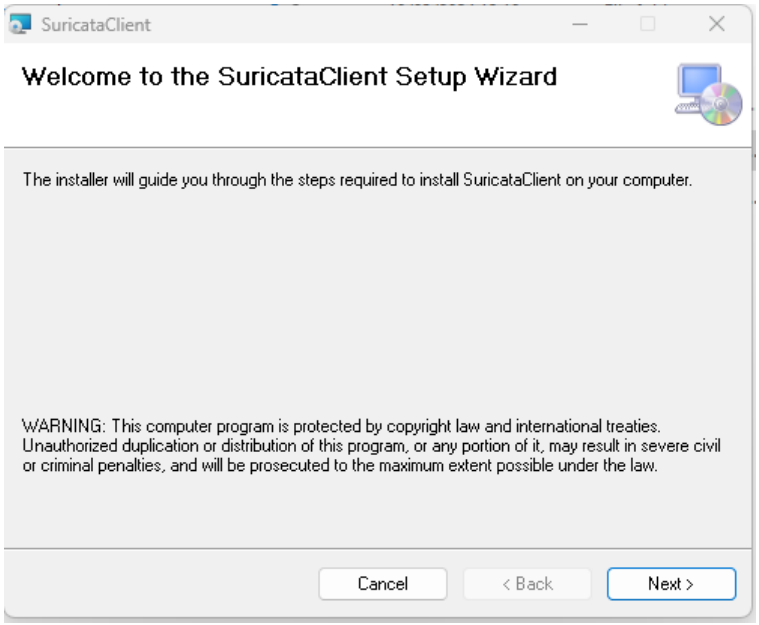


3. The notify message will appear. Click **Yes**.

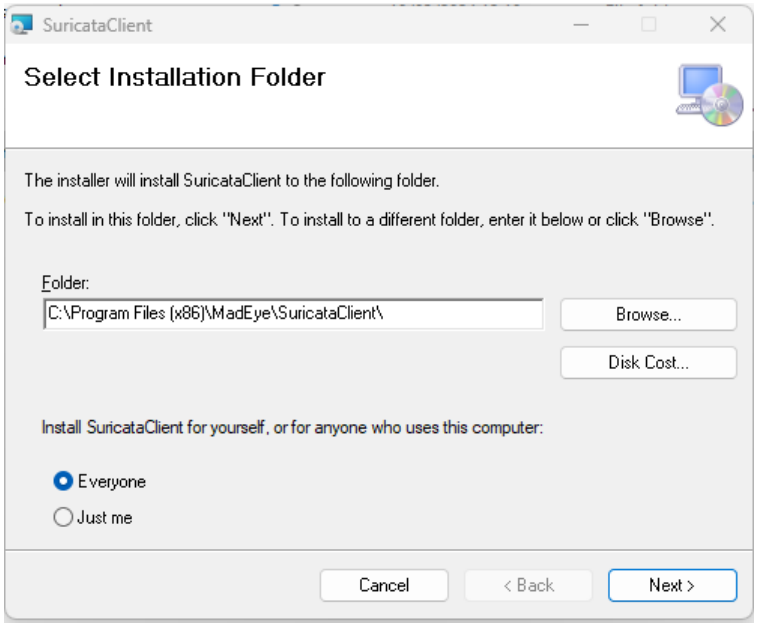
4. The confirmation message will appear. Click **Y** and then click **Enter.**



5. Once the uninstallation is complete, SuricataClient Setup Wizard will open. Click **Next**.

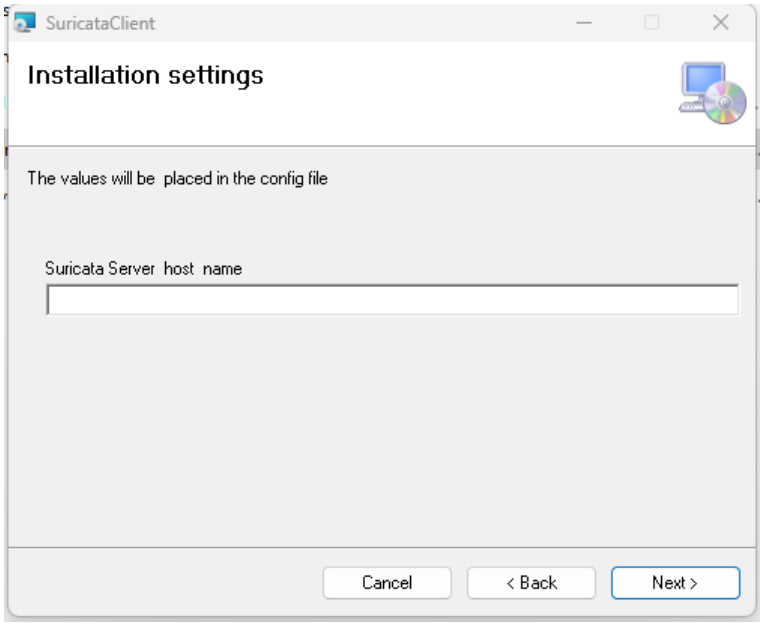


6. The installer will suggest installing Suricata Client in a folder on the C drive for all users. Click **Next** to confirm or adjust the default selections and then click **Next**.

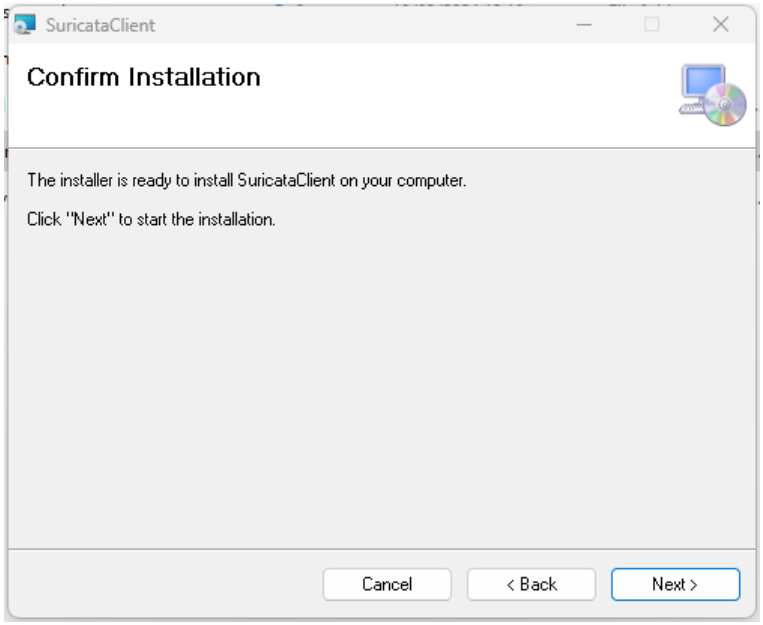


7. Enter the **Suricata Server host name** and click **Next**.

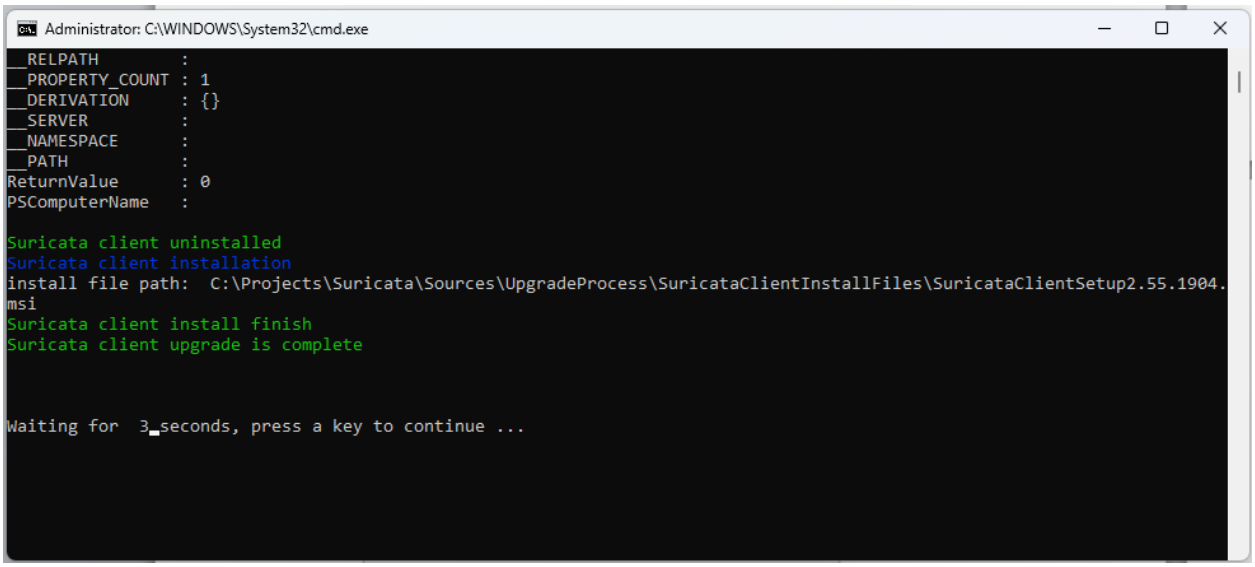
> **Note:**
>
> This is the name of the device Suricata Server was installed on in the previous section. To find the name of the Windows device you are currently on, search for "About" from the Windows Start menu. The name is displayed under "Device name".



8. Click **Next** to install Suricata Client.



9. Once the installation is complete, the command window will close itself after 10 seconds

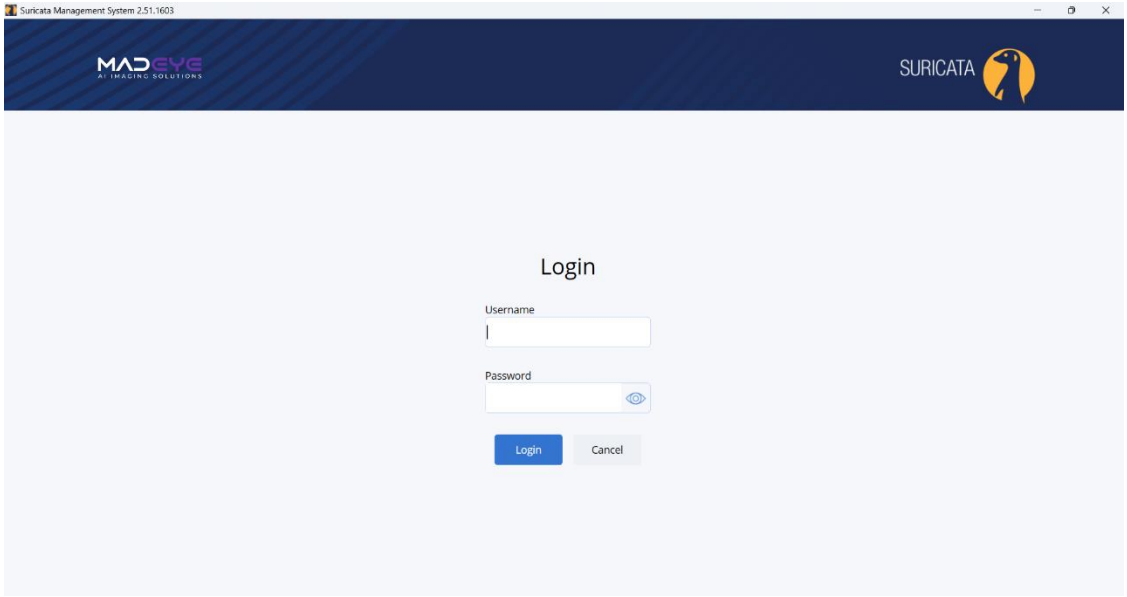


10. A shortcut to Suricata Client will be saved to the desktop. Double click to open the application.

11. At the login screen, enter the following credentials:

    Username: sa

    Password: sa



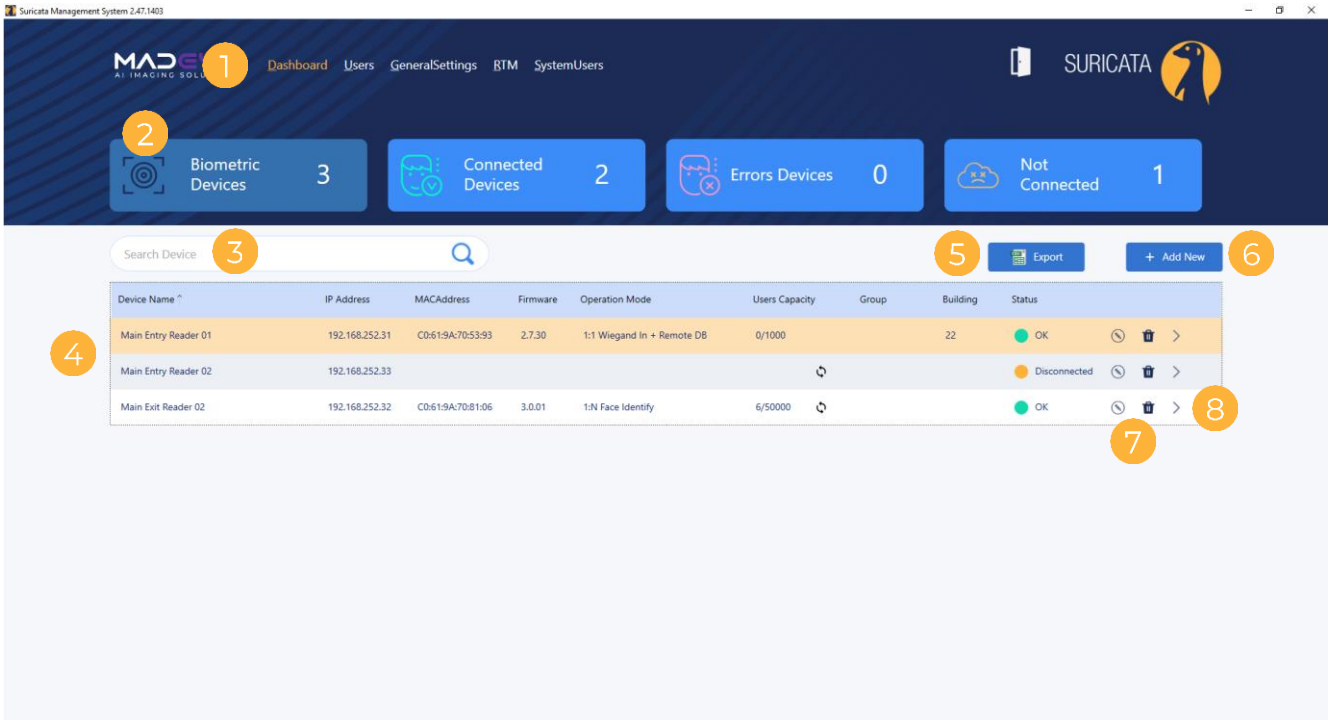
> **Note:**
>
> These credentials are intended to be used for initial setup only and should be changed or deleted to ensure your system remains secure. Refer to System Users.

12. Once logged in, refer to the other sections of this guide to:

    - understand the Dashboard and configure devices
    - add Users
    - configure General Settings
    - configure Real-Time Monitoring (RTM)
    - add System Users

Dashboard

Once logged in to Suricata, the Dashboard will be displayed. Refer to the table below for explanations.



1 **Menu**
Click to navigate between Dashboard, Users, General Settings, Real-Time Monitoring (RTM), and System Users.

2 **Device Status Filters**
Provides a summary of device statuses (all devices, connected, errors, not connected) and acts as a filter for the table below. By default, the **Biometric Devices** (total devices) filter is selected. Click to filter by another status.

3 **Device Search**
Search by Device Name or IP Address. Enter a search term and press Enter to view results.

4 **Device Table**
Lists all matching results for the selected filter or search. Orange indicates selection. Click on a heading to sort data. Click again to reverse the sort order.

5 **Export**
Click to download a CSV file of the current table view (i.e., if a filter is applied, only the filtered results will be exported).

6 **Add New**
Refer to Add a New Device.

7 **Edit/Delete**
Refer to Edit or Delete a Device

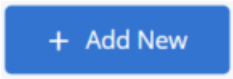
8 **Device Settings**
Refer to

Configure Device Settings.

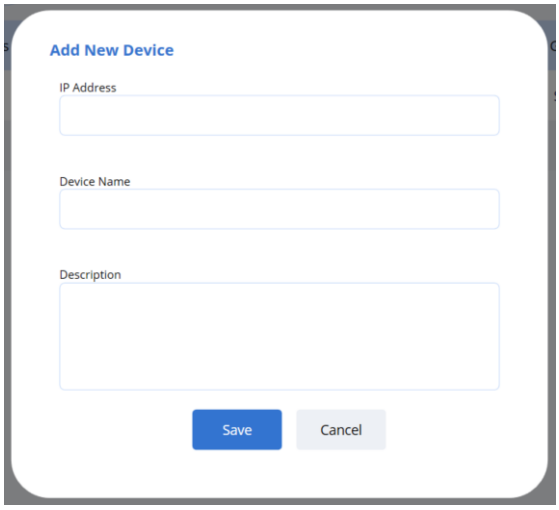
Add a New Device

To add a new biometric device to the system:

1. Go to the **Dashboard** and click **+ Add New**.



2. Enter:
   - The **IP Address** (required)
   - The **Device Name** (required)
   - A **Description** (optional)



3. Click **Save** to add the new device.

Note:

Once the device has been added, click the arrow to configure the device settings (refer to Configure Device Settings).

The **Sync Users** icon will appear if the device's Operation Mode is set to 1:N Face Identify, 1:1 Wiegand In + Local DB, 1:N Face Identify - Remote DB – Server, or 1:1 DESFire Verify + Remote DB. Click the icon to sync users.

Edit or Delete a Device

To edit the **IP Address**, **Device Name**, or **Description** a biometric device:

1. Go to the **Dashboard** and click the **Edit** icon to the right-hand side of the device.

2. The Edit Device box will appear. Edit the details as required.
3. Click **Save**.

To delete a biometric device:

1. Go to the **Dashboard** and click the **Delete** icon to the right-hand side of the device.

2. Click **Delete** to confirm.

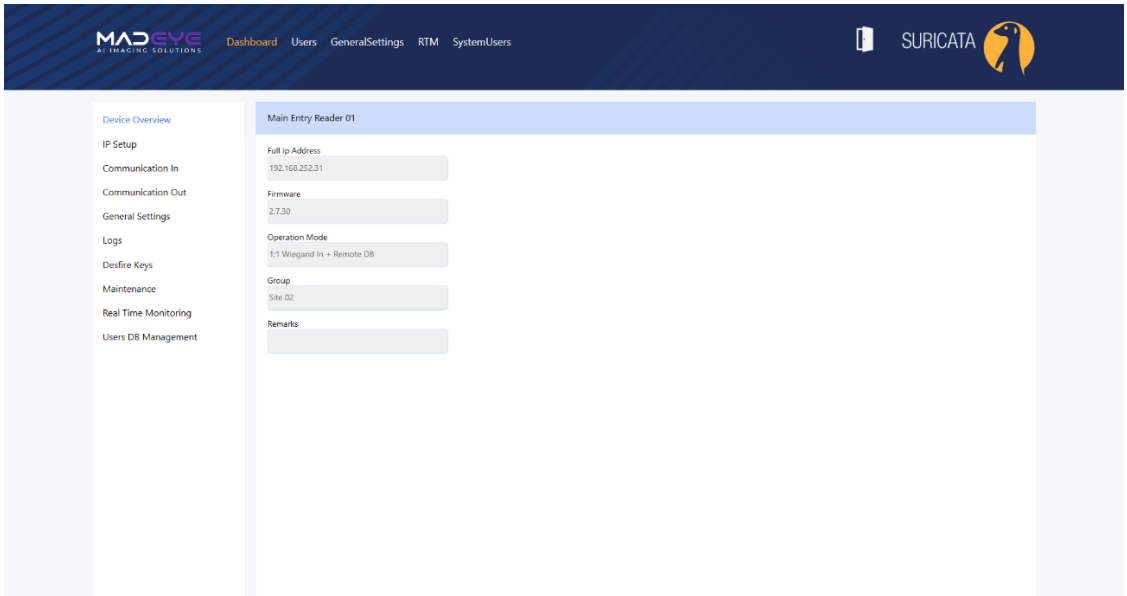
> **Warning:**
>
> Deleting a device cannot be undone.

Configure Device Settings

To view and configure device settings:

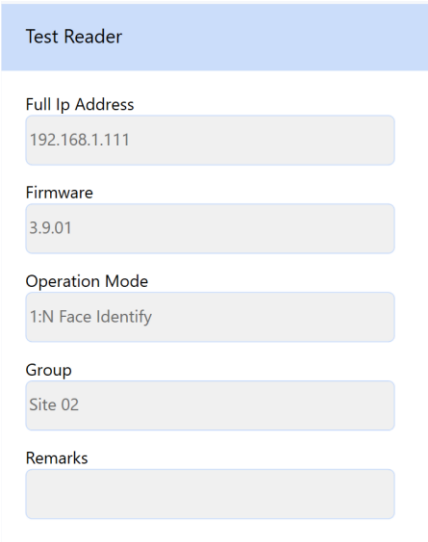
1. Go to the **Dashboard** and click the arrow to the right-hand side of the device.

2. The **Device Overview** page will be displayed. Use the menu on the right-hand side to navigate the device settings. Refer to the sections below for default and recommend settings.



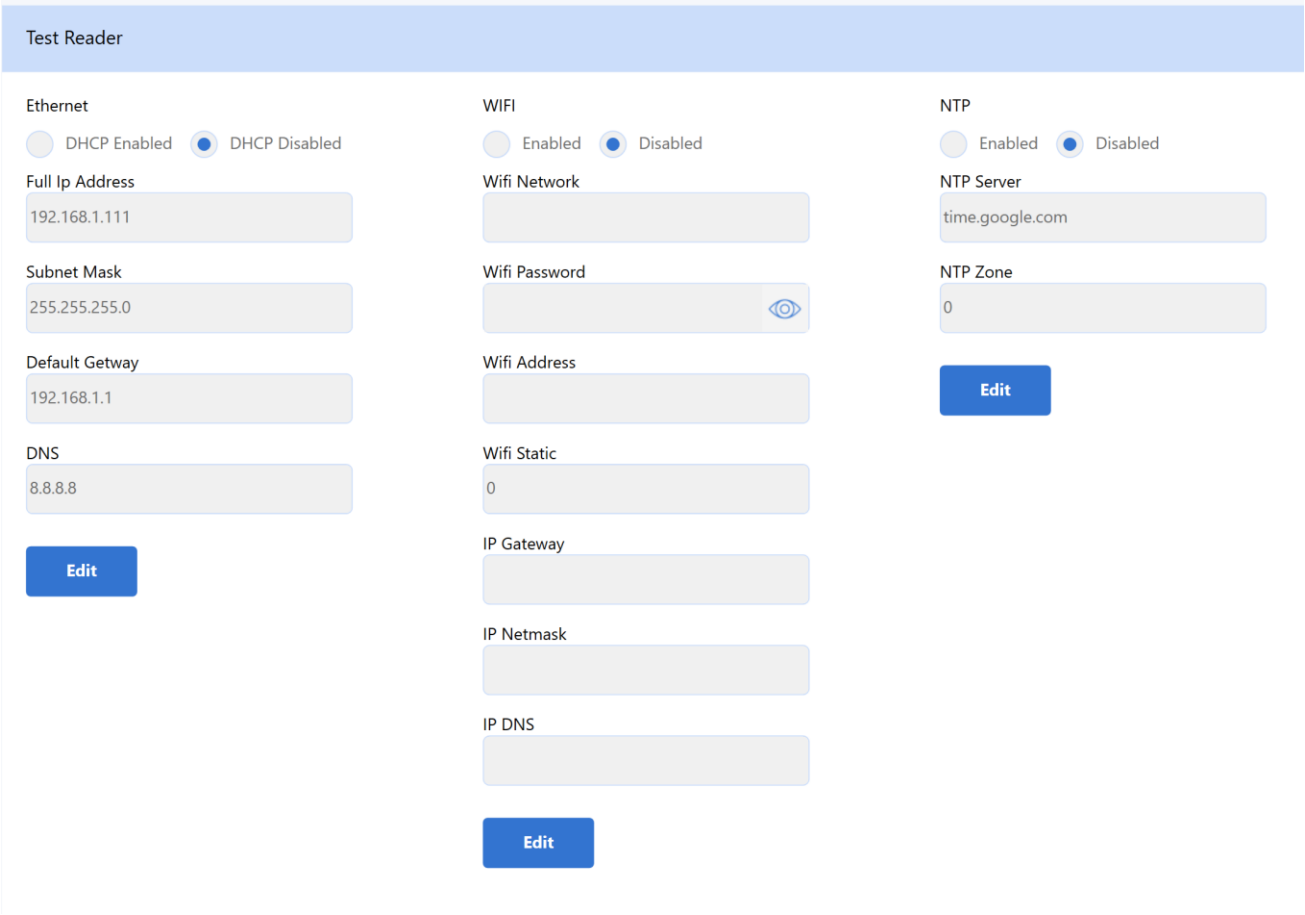
Device Overview

This page summaries the basic device information.



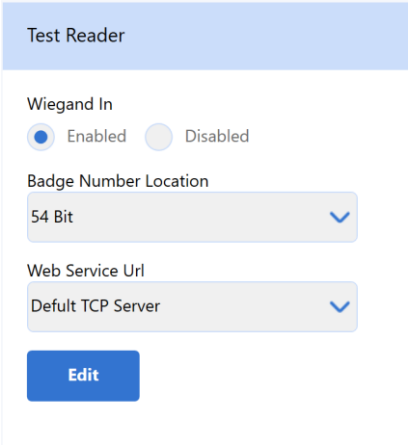
IP Setup

Use this page to configure the device's ETH0 (ethernet) and WLAN (wirless) network interfaces.



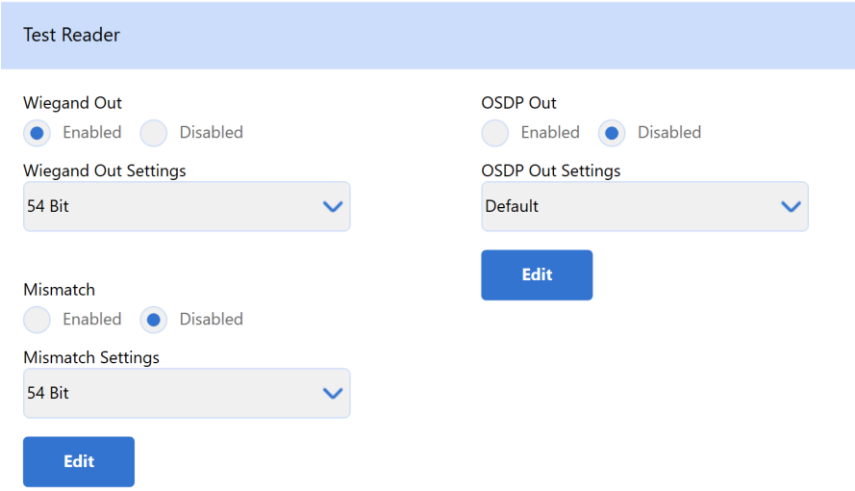
Communication In

Use this page to configure the device's Wiegand In settings. All protocols available for configuration are preset under General Settings.



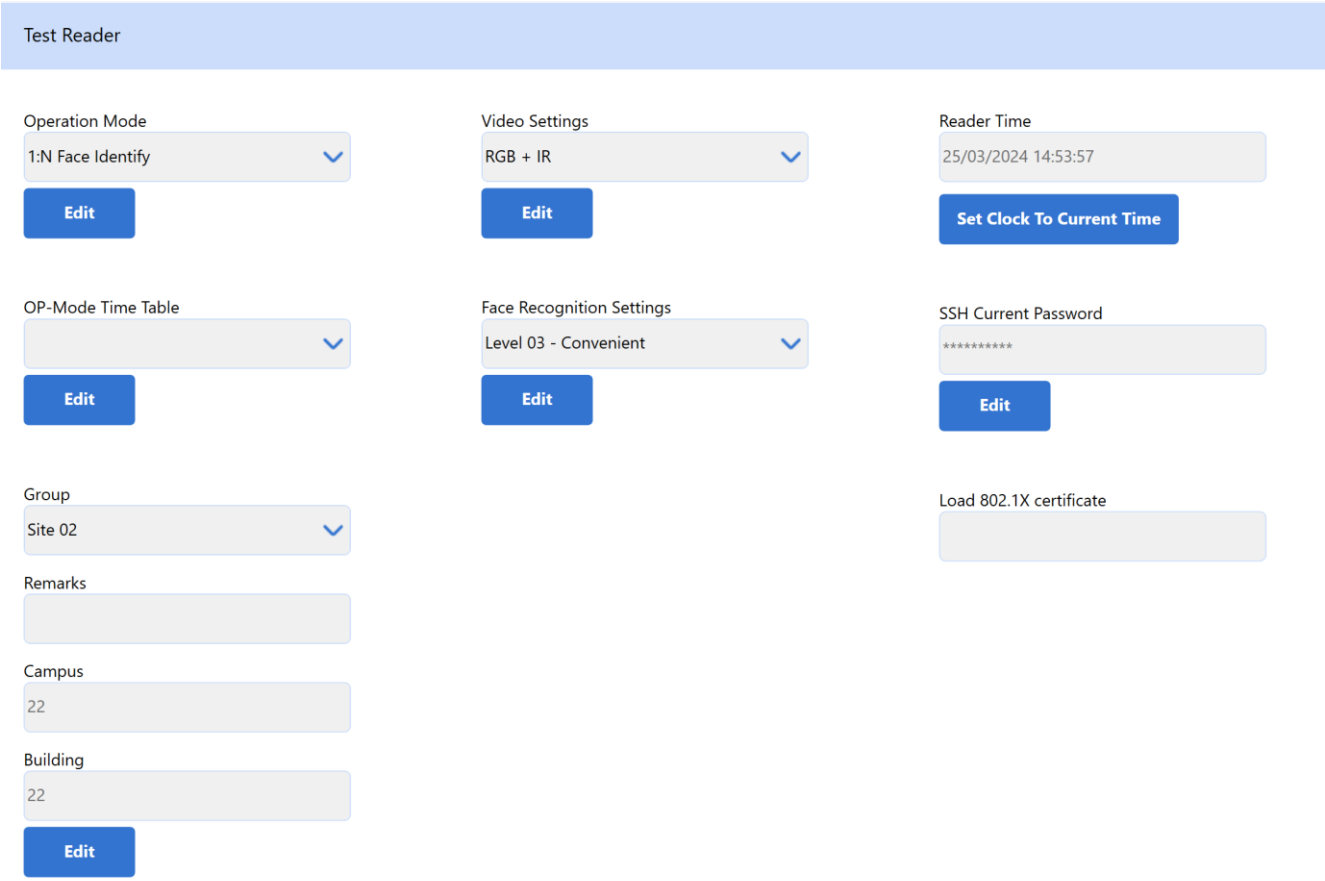
Communication Out

Use this page to configure the device's Wiegand OUT/OSDP settings. All protocols available for configuration are preset under General Settings.



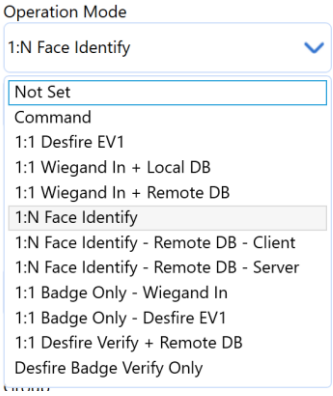
General Settings

Use this page to configure the device's general settings (including Operation Mode, Group, Building, Video Settings, and Time).



Field	Description	Default	Recommended	Range
Operation Mode	1:1/1:N/TOC/SER MODE			
Video Settings	Camera settings	NIR		

Select the device Operation Mode from the drop-down menu.

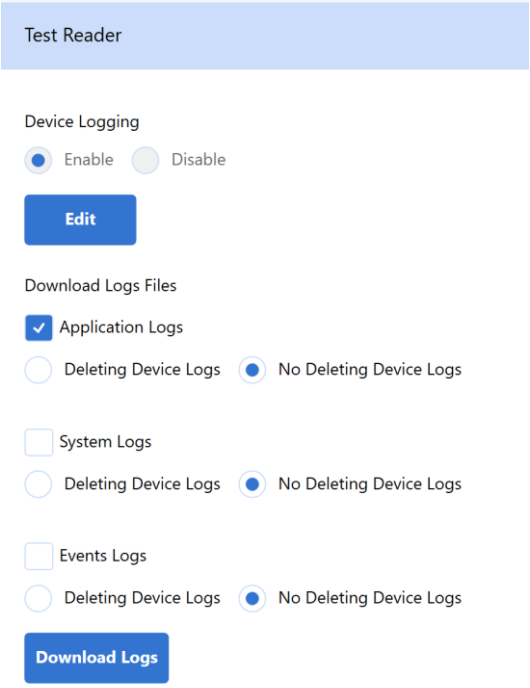


Mode	Description
Command	The device waits for commands input. This mode is recommended for units used for templates extraction (enrollment).
1:1 DESFire EV1	The device waits for a DESFire card with biometric data stored on it. Once data retrieved, the device will start face verification.
1:1 Wiegand In + Local DB	The device waits for userID input on Wiegand interface. If UID exists on local DB, the device will start face verification.
1:1 Wiegand In + Remote DB	The device waits for userID input on Wiegand interface. Once retrieved, the UID is sent to server for biometric data request. Once biometric data is retrieved from server, the device will start face verification.
1:N Face Identify - Device	The device is in face capture mode. Each face captured will be authenticated against stored biometric data on the device (up to 100,000 users).
1:N Face Identify - Remote DB - Client	The device is in face capture mode. Each face captured will be sent to “server unit” for identification.
1:N Face Identify - Remote DB - Server	The device waits for face templates captured and extracted by “client units”.
1:1 Badge Only - Wiegand In	Badge only credential. The device waits for userID input on Wiegand in interface.
1:1 Badge Only – DESFire EV1	Badge only credential. The device waits for userID input on internal card reader.
1:1 DESFire Verify + Remote DB	
DESFire Badge	

Mode	Description
Verify Only	

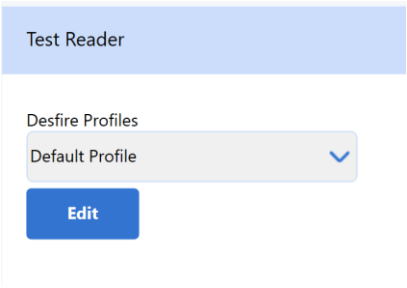
Logs

Use this page to configure log settings and download logs for this device.



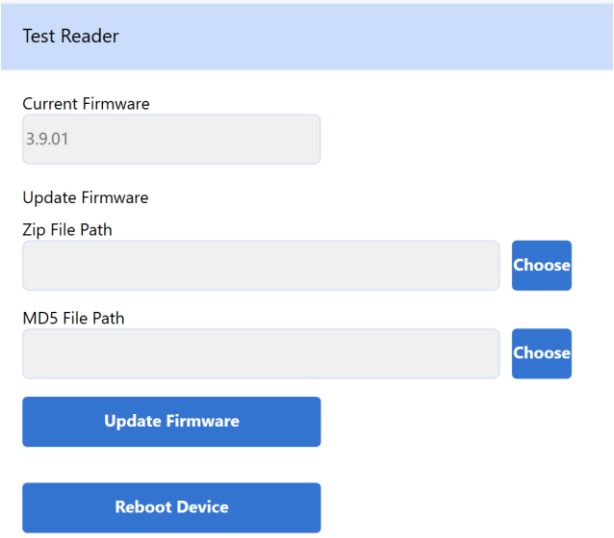
DESFire Keys

Use this page to enable DESFire card formats preset in General Settings.



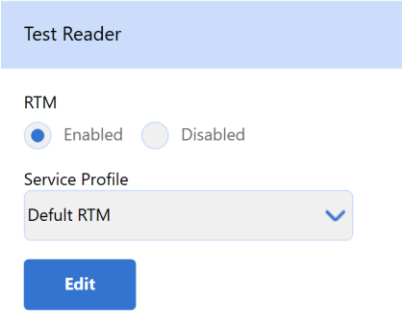
Maintenance

Use this page to update the device's firmware.



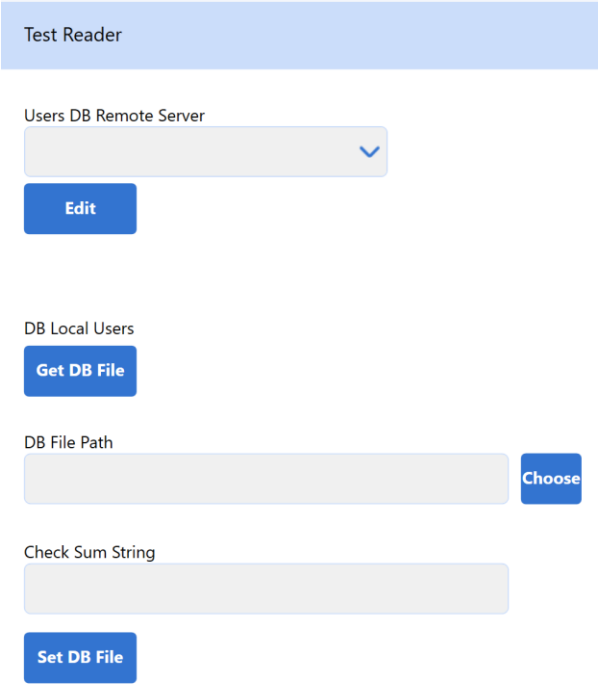
Real Time Monitoring

Use this page to enable/disable real time monitoring (RTM) profiles preset in General Settings.



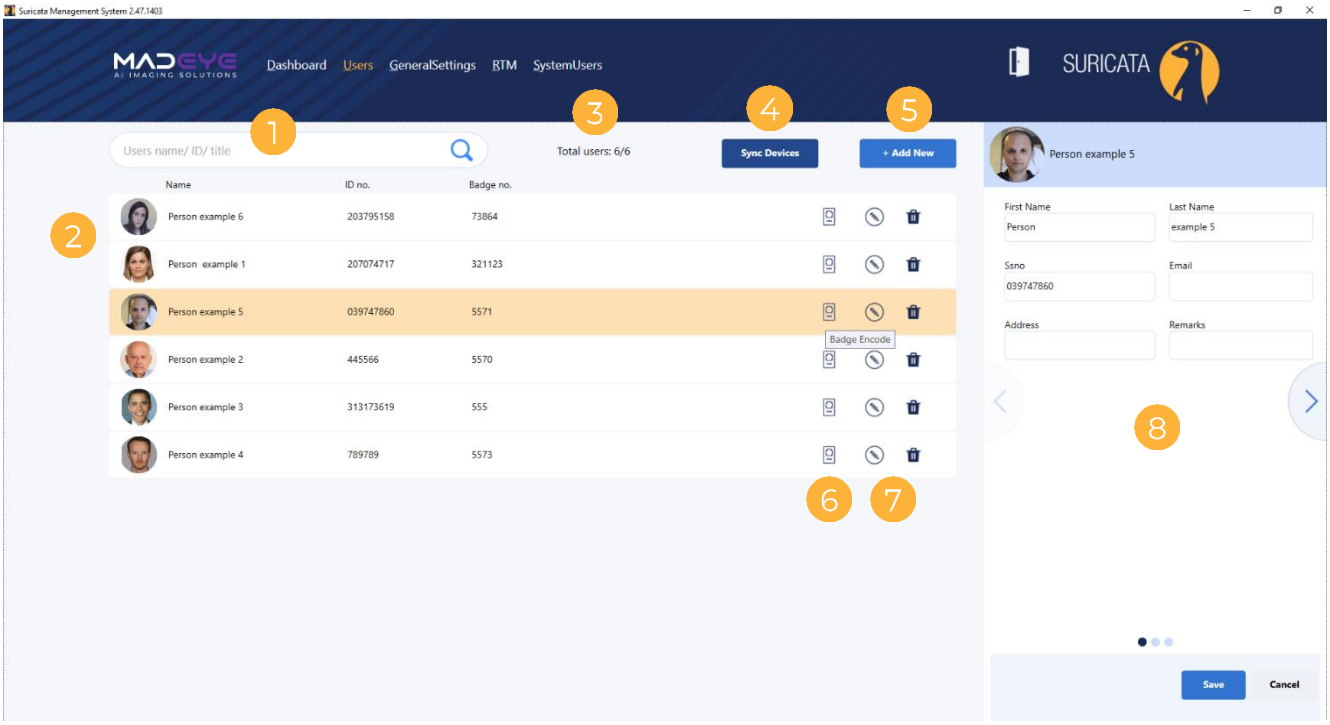
Users DB Management

Use this page to configure setting/getting the users DB on the device.



Users

From the Users page, you can manage the database of cardholders and their biometric data that the system will compare when a User requests entry.



1 **User Search**
Search Users by Name, Ssno, or Badge number. Enter a search term and press Enter to view results.

2 **Users Table**
Lists all Users by default or matching search results. Orange indicates selection.

3 **Total Users**
Displays the total number of Users in the database.

4 **Sync Devices**
Click to sync all biometric devices.

5 **Add New**
Refer to Add a New User.

6 **Badge Encode**
Refer to Encode a User's Badge.

7 **Edit/Delete**
Refer to Edit or Delete a User

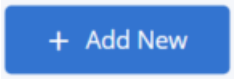
8 **Details Window**
User the navigation arrows to view the details of a selected User.

Add a New User

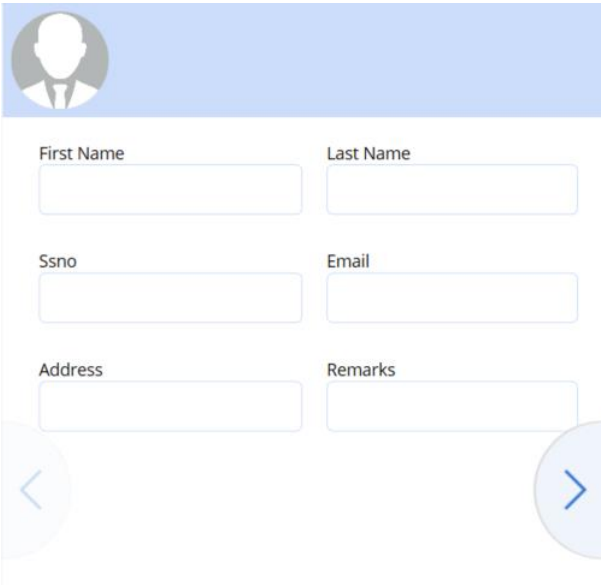
> **Note:**
> At least one biometric device needs to be connected and enrolled to be able to add a new user and verify the image.

To add a new User/cardholder:

1. Go to **Users** and click **+ Add New**.



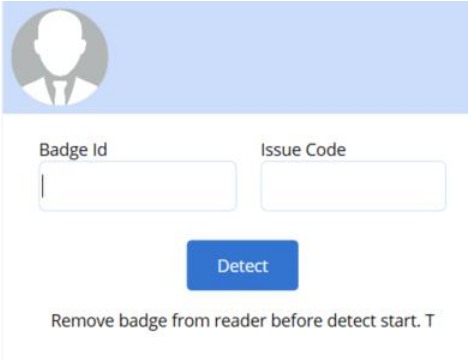
2. Enter:
   - The User's **First Name** (required)
   - The User's **Last Name** (required)
   - The User's **Ssno** (required)
   - The User's **Email** (optional)
   - The User's **Address** (optional)
   - Any **Remarks** (optional)



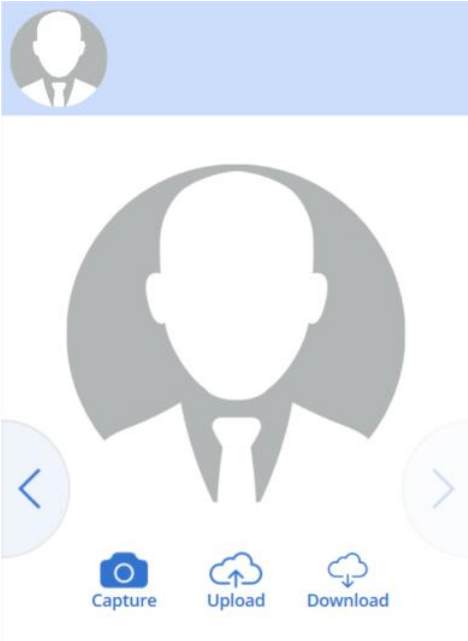
3. Click **Save** or click the navigation arrow on the right-hand side to continue.

4. Click **Detect** and follow the instructions on screen to scan the User's badge or manually enter:

    - The User's **Badge Id** (required)
    - The Badge Id **Issue Code** (optional)



5. Click **Save** or click the navigation arrow on the right-hand side to continue.
6. Click **Capture** to take a photo of the User using your computer's webcam or click **Upload** to upload an existing image.



7. Click **Image Verify** to confirm the image sufficiently matches the User's biometric data.

8. Click **Save**.

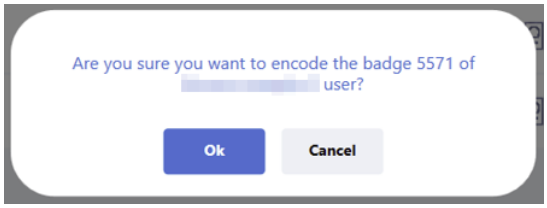
Encode a User's Badge

> **Note:**
>
> The Badge encode option is only available when the Suricata Client machine is defined in DESFire Encoder Settings.

To encode a User's badge:

1. Go to the **Users** page and click the **Badge Encode** icon to the right-hand side of the User.

2. Click **Ok** to encode.



Edit or Delete a User

To edit a User:

1. Go to the **Users** page and click the **Edit** icon to the right-hand side of the User.

2. The selected User will appear in the Details Window. Edit any of the User's details, and click the navigation arrow on the right-hand side to edit the Users' **Badge Id** and **Issue Code**, and click again to edit the User's image.
3. Click **Save**.

To delete a User:

1. Go to the **Users** page and click the **Delete** icon to the right-hand side of the User.

2. Click **Delete** to confirm.

> **Warning:**
>
> Deleting a User cannot be undone.

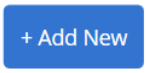
General Settings

From the General Settings page, you can configure existing Profiles/Settings and add new Profiles/Settings.

Add a New Profile/Setting

To add a new Profile/Setting:

1. Go to the **General Settings** page and select the Profiles/Settings type on the left-hand side.
2. Click **+ Add New**.



3. Click Save.

Edit or Delete a Profile/Setting

To edit a Profile/Setting:

1. Go to the **General Settings** page and select the Profile/Setting.
2. Click the **Edit** icon to the right-hand side of the Profile/Setting.

3. Edit the Profile/Setting as required.
4. Click **Save**.

To delete a Profile/Setting:

1. Go to the **General Settings** page and select the Profile/Setting.
2. Click the **Delete** icon to the right-hand side of the Profile/Setting.

3. Click **Delete** to confirm.

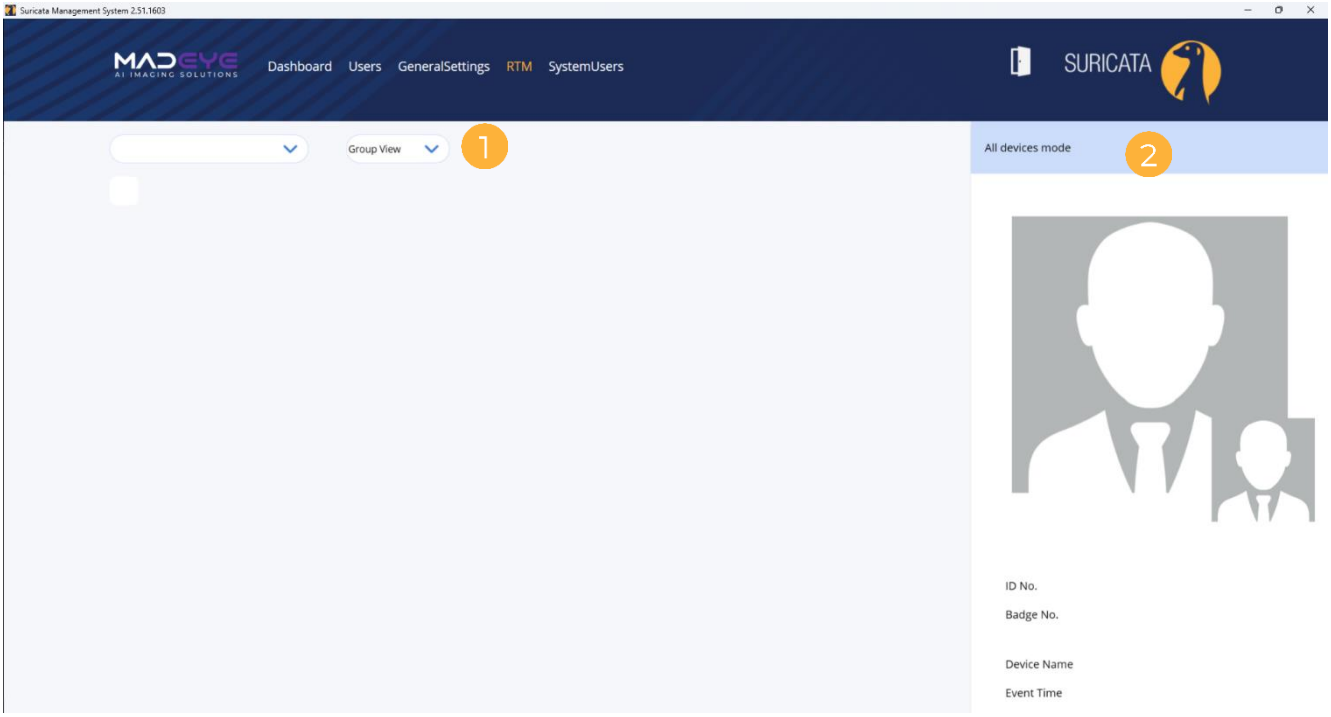
Warning:

Deleting a Profile/Setting cannot be undone.

Real-Time Monitoring (RTM)

From the Real-Time Monitoring (RTM) page, you can monitor events from online devices.

The system will present the database (DB) image and real-time image of identified users or just the real-time image in case of mismatch.

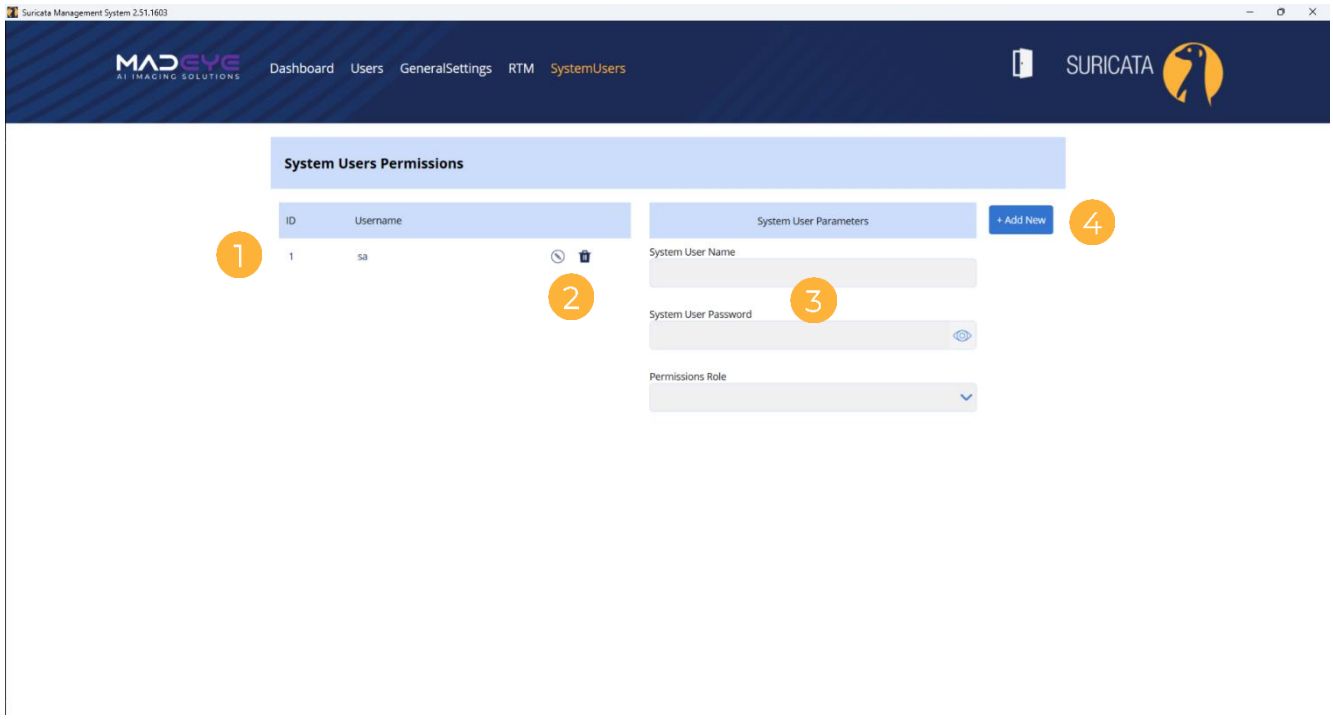


1 **View Menu**
Switch between **Group View** and **Single View**.

2 **Details Window**
Displays the details for the selected User.

System Users

From the System Users page, you can manage access to Suricata Client.



1. **System Users Table**
   Lists all System Users. Orange indicates selection. Click on a heading to sort data. Click again to reverse the sort order.

2. **Edit/Delete**
   Refer to Edit or Delete a System User.

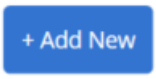
3. **System User Parameters**
   Displays the details for the selected System User or new System User.

4. **Add New**
   Refer to Add a New System User.

Add a New System User

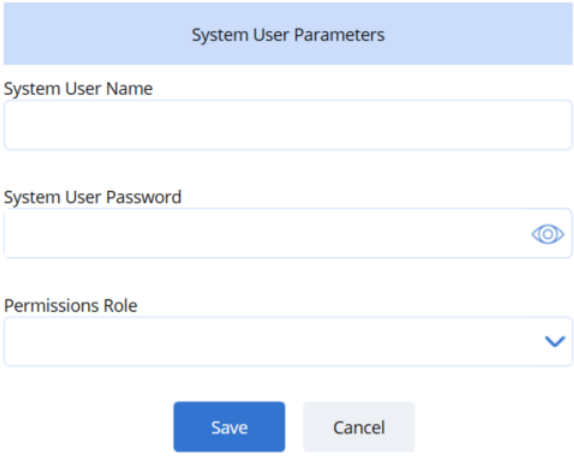
To add a new System User:

1. Go to **System Users** and click **+ Add New**.

2. Enter/select the new System User's parameters:
   - Enter the **System User's Name** (required)
   - Enter the **System User's Password** (required)
   - Select the **Permissions Role** from the drop-down menu (required)

> **Note:**
>
> Refer to System Roles for an explanation of each role.



3. Click **Save**.

Edit or Delete a System User

To edit a System User:

1. Go to the **System Users** page and click the **Edit** icon to the right-hand side of the System User.

2. The selected System User will appear under System User Parameters. Edit any of the System User's details and click **Save**.

> **Tip:**
>
> Use the **View Password** icon to view the current password.
>
> 

To delete a System User:

1. Go to the **System Users** page and click the **Delete** icon to the right-hand side of the System User.

2. Click **Delete** to confirm.

> **Warning:**
>
> Deleting a System User cannot be undone.

System Roles

Permissions for System Users are role-based. Refer to the table below for an explanation of what permissions are granted to each role:

Role	Dashboard /Devices	Users	General Settings	RTM	System Users
Admin	☑	☑	☑	☑	☑
ConfigurationManager	☑	-	-	☑	-
CardholderManager	-	☑	-	-	-
RTMView	-	-	-	☑	-

API

This section describes the API functions available and DTO schemas used by API functions.

General

- Suricata has been developed in .NET framework 4.5.
- Suricata Server must be installed to use the API.
- The API is based on the SignalR software library for sending two-way server-client messages.
- The Suricata Server uses the Microsoft.AspNet.SignalR server library.
- API clients must use the Microsoft.AspNet.SignalR.Client (version 2.4.3) library to connect to the Suricata Server API.
- API clients can call all API functions (refer to API Functions) and register a callback for events (refer to API Events for Callback Registration).

Connection Details

Parameter	Explanation
ServerName	The name of the device where Suricata Server is installed/running
Port	7082
Protocol	http
HubName	SuricataApiHub

API Functions

Parameter	Explanation
UpdateCardholder(CardholderData personDataObj, string interfaceSource)	Add (if not exists) or update (if exists) user in Suricata and load to all online devices in Face operation mode. Ssno parameter used as key value for check if user exists. CardholderData schema described in classes schema section. Manadatory fields are: Ssno, BadgeId, Image If FaceTemplate is not provided in personDataObj, it will be created by ExtractFaceTemplate function for create template from provided Image interfaceSource – source system name (Lenel, Morpho, etc)

Parameter	Explanation
ExtractionResult ExtractFaceTemplate(byte[] image, int empId)	Image – provided for create biometric face template empId – provided for referencing on callback event
DeleteCardholder(string ssno)	Delete user by ssno number from Suricata and all online devices
DeleteCardholderByBadge(long badgeId)	Delete user by badge number from Suricata and all online devices
GetCardholder(string ssno)	Get user by ssno number from Suricata
UpdateCardholderInfo(CardholderData personDataObj, string interfaceSource)	Update (if exists) user's info (without image and biometric face template) in Suricata and all online devices in Face operation mode. Ssno parameter used for check if user exists. CardholderData schema described in section 5, manadatory fields are: Ssno, BadgeId interfaceSource – source system name (Lenel, Morpho, etc)
GetLastSuccessRTMEvents(string ipAddress, int lastEvents)	Retrieve last success (success user identification) RTM events ipAddress – device address where events were arrived lastEvents – number of last events will be returned on response
GetLastRTMEvents(string ipAddress, int lastEvents)	Retrieve last (success user identification and biometric mismatch) RTM events ipAddress – device address where events were arrived lastEvents – number of last events will be returned on response
OnRTMEventSubscribe	Subscribe to Suricata Server to receive RTM events

API Events for Callback Registration

Parameter
IHubProxy.On<string>("UpdateCardholderResult", (resultMessage))
IHubProxy.On<byte[], int, string>("ExtractFaceTemplateResult", (template, empId, status))
IHubProxy.On<string>("DeleteCardholderResult", (resultMessage))
IHubProxy.On<string>("DeleteCardholderByBadgeResult", (resultMessage))
IHubProxy. On<CardholderRequestResult>("GetCardholderResult", (result))

Parameter
IHubProxy.On<CardholderRequestResult>("UpdateCardholderInfoResult", (result))
IHubProxy.On<RTMEventsResult>("GetLastSuccessRTMEventsResult", (result))
IHubProxy.On<RTMEventsResult>("GetLastRTMEventsResult", (result))
IHubProxy.On<PersonData, string, string, string, string, byte[]>("RTMEventArrived", (person, accessStatus, readerIp, readerName, time, imageByteArray))

Schema Classes

```
class CardholderData
{
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Ssno { get; set; }
        public string Address { get; set; }
        public string Email { get; set; }
        public string Remarks { get; set; }
        public Nullable<long> BadgeId { get; set; }
        public Nullable<long> IssueCode { get; set; }
        public byte[] Image { get; set; }
        public byte[] FaceTemplate { get; set; }
}

class ExtractionResult
{
        public byte[] Template { get; set; }
        public int Status { get; set; }
        public string ErrorMessage { get; set; }
}

class CardholderRequestResult
{
       public CardholderData Cardholder { get; set; }
       public int Status { get; set; }
       public string ErrorMessage { get; set; }
}

class RTMEventsResult
{
       public List<RTMEventsExt> rtmEventsExts { get; set; }
       public string Status { get; set; }
       public string ErrorMessage { get; set; }
}

class RTMEvents
{
       public long ID { get; set; }
       public string EventDateTme { get; set; }
       public Nullable<long> BadgeId { get; set; }
       public string ReaderIp { get; set; }
}
```

```
class RTMEventsExt : RTMEvents
{
        public string Ssno { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string BiometricIdentificationResult { get; set; }
}

class PersonData
{
        public long PId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Ssno { get; set; }
        public string Address { get; set; }
        public string Email { get; set; }
        public string Remarks { get; set; }
        public long? BadgeId { get; set; }
        public long? IssueCode { get; set; }
        public byte[] Image { get; set; }
        public long? PersonBiometricTemplateId { get; set; }
}
```

Code Samples

Connection to Suricata API

string protocol = “http”;

string serverName = “SuricataServerName”;

string serverPort = “7082”;

string hubName = “SuricataApiHub”;

```
Microsoft.AspNet.SignalR.Client.HubConnection hubConnection = new
Microsoft.AspNet.SignalR.Client.HubConnection(protocol + "://" + serverName + ":" + serverPort);

Microsoft.AspNet.SignalR.Client.IHubProxy proxy = hubConnection.CreateHubProxy(hubName);

hubConnection.Start().Wait();
```

Call API function

```
await proxy.Invoke("GetCardholder", ssno);
```

Registration to async result

```
proxy.On<CardholderRequestResult>("GetCardholderResult", (result) =>
        {
          //GetCardholderResult callback from server after call GetCardholder method.
        });
```

Logs

By default, log files are saved in C:\ProgramData\MedEye\SuricataServer\Logs.

Logs include details such as client connection to API hub status, start/finish call API function and additional details of API function process.

SURICATA