Aigoeye

Enabling Smart Surveillance

Milestone Integration Plugin - SarvAI

User Manual

Aigeeye
Enabling Smart Surveillance

TABLE OF CONTENTS

Introduction

In the realm of modern security and surveillance, the rise of technology has brought forth numerous benefits, but it has also introduced new challenges. One significant challenge faced by the users of Milestone surveillance systems is the frequent occurrence of unjustified alarms(False alerts), which not only disrupt daily operations but also strain the time and resources of surveillance system. In response to this challenge, we present "Aigoeye," a groundbreaking solution which integrates with Milestone's eco system to enable a new era of smart surveillance.

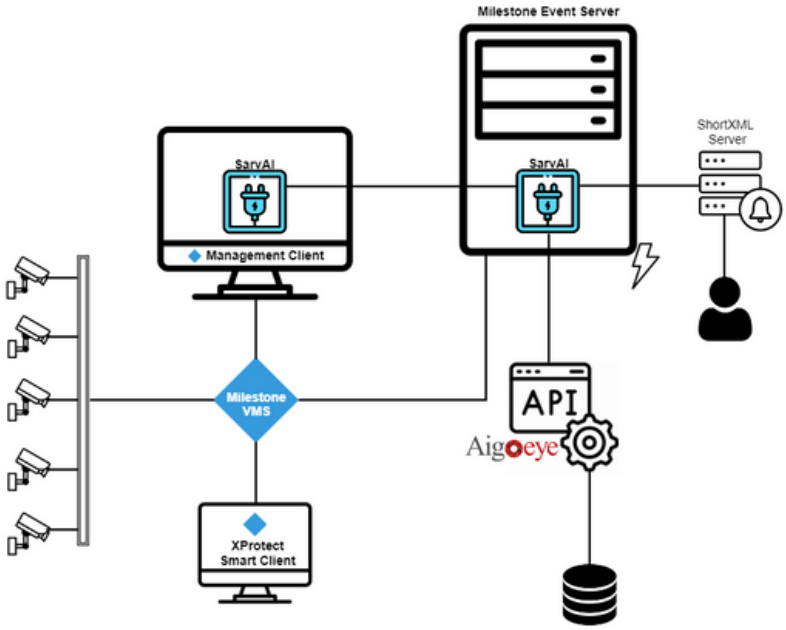
In the context of large-scale enterprises, imagine a company equipped with a network of 100 cameras which uses Milestone as their surveillance solution. They have activated 'Motion Detection' to generate alarms in response to potential human interventions within their surveillance range. The consequence of this setting can often be overwhelming for the security teams. They get flooded with a lot of alarms, sometimes more than thousand alarms in a single day. Regrettably, the actual incidents of true human intervention often prove to be exceedingly rare when compared to the false alarms.

Aigoeye's AI solution lies in its ability to discern real human intervention threats from false alarms. This AI solution not only enhances security but also streamlines the resource allocation process, allowing personnel to concentrate their efforts where they are most needed, ultimately resulting in a more efficient and cost-effective approach to security and surveillance management.

How it Works

Our innovative solution comprises two integral components, each playing a crucial role in revolutionizing the way we handle surveillance alarms. The first component, known as "*SarvAI*," is a dynamic plugin designed to integrate with the Milestone system. SarvAI's primary function is to extract information of alarms generated by Milestone's event server and facilitate the exportation of the corresponding alarm videos to our robust API, the second component of our solution, "*Aigoeye API*."

The Aigoeye API serves as the brain behind our operation, receiving and meticulously analyzing the incoming alarm videos. It then sends a response back to the SarvAI Plugin. Based on the response received from the Aigoeye API, the SarvAI Plugin takes action, dynamically updating the alarm state as either "*New*" or "*Closed*." Simultaneously, it generates and dispatches a ShortXML message to the relevant client, notifying them of any required human intervention within the surveillance range.



Before Installing

“To facilitate the seamless functioning of our solution, it is imperative to begin by installing the Aigoeye API.”

Requirements

Milestone XProtect

Milestone XProtect Corporate 2021 R1, R2, 2022 R1, R2, 2023 R1

Plugin

SarvAI 1.0.1

API

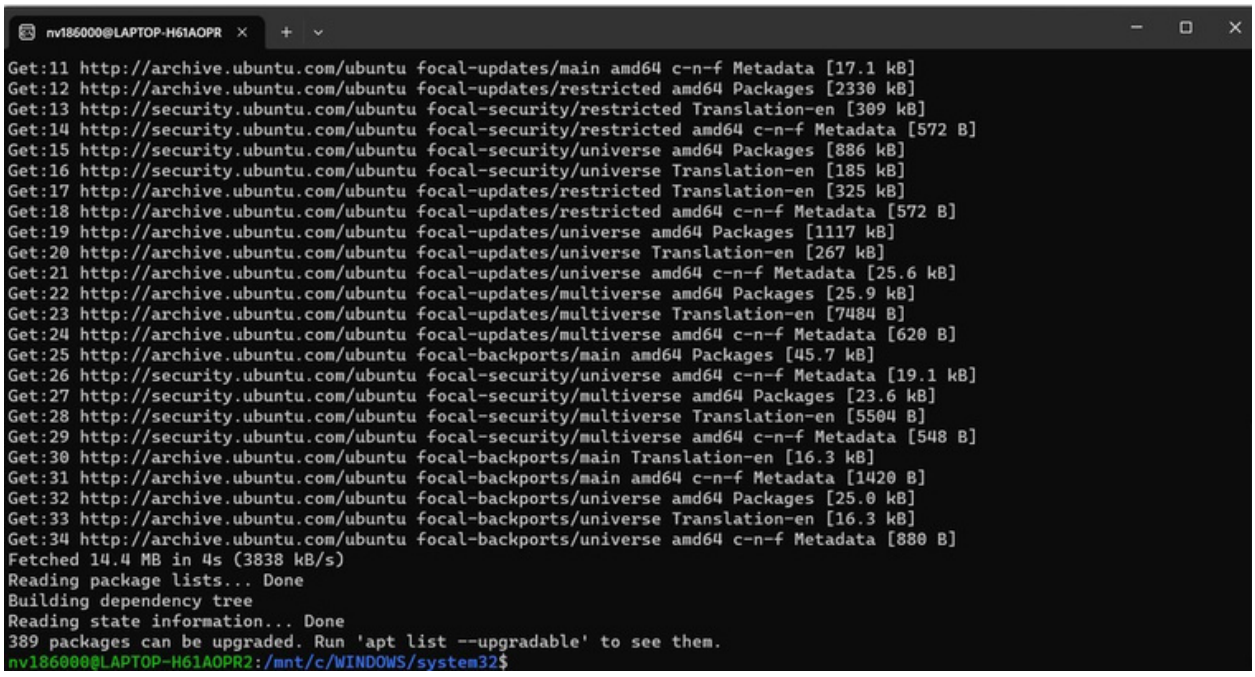
Linux OS version 22.04
Python 3.7
MySQL 8.0
nginx

API Installation

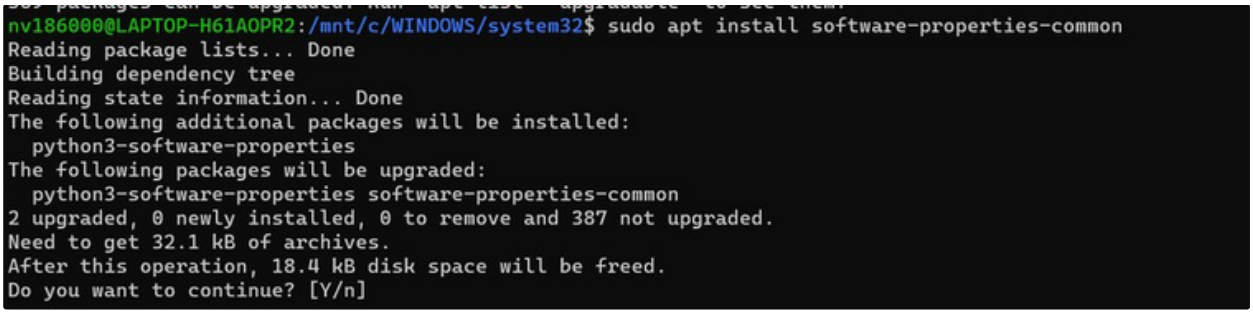
1. Install Linux OS version 22.04 into system and connect from putty
2. Run following command to install python 3.7 and pip

```
sudo apt update

sudo apt install software-properties-common
```




```
sudo apt install software-properties-common
```




```
sudo add-apt-repository ppa:deadsnakes/ppa
```

```
ubuntu@ip-172-31-38-102:~$ sudo add-apt-repository ppa:deadsnakes/ppa
Repository: 'deb https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu/ jammy main'
Description:
This PPA contains more recent Python versions packaged for Ubuntu.

Disclaimer: there's no guarantee of timely updates in case of security problems or other issues.
n server), you do so at your own risk.

Update Note
===========
Please use this repository instead of ppa:fkrull/deadsnakes.
```

```
sudo apt install python3.7
```

```
ubuntu@ip-172-31-38-102:~$ sudo apt install python3.7
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bzip2 libpython3.7-minimal libpython3.7-stdlib mailcap mime-support python3.7-minimal
Suggested packages:
  bzip2-doc python3.7-venv binfmt-support
The following NEW packages will be installed:
  bzip2 libpython3.7-minimal libpython3.7-stdlib mailcap mime-support python3.7 python3.7-minimal
0 upgraded, 7 newly installed, 0 to remove and 129 not upgraded.
Need to get 4733 kB of archives.
After this operation, 17.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
sudo apt install python3-pip
```

```
ubuntu@ip-172-31-38-102:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6 libc6-
  libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgcc-s1 libgd3 libgomp1 libisl23 libitm1 libjbig0 libjp
  libmpc3 libnsl-dev libpython3-dev libpython3.10 libpython3.10-dev libpython3.10-minimal libpython3.10-stdlib
  libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-wheel pytho
Suggested packages:
  cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf automake
  glibc-doc bzr libgd-tools libstdc++-11-doc make-doc python3.10-venv python3.10-doc binfmt-support
Recommended packages:
  libnss-nis libnss-nisplus
The following NEW packages will be installed:
  build-essential cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11 gcc-11
  libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libdef
  libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery l
  libpython3.10-dev libquadmath0 libstdc++-11-dev libtiff5 libtirpc-dev libtsan0 libubsan1 libwebp7 libxpm4 lin
  python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
The following packages will be upgraded:
  gcc-12-base libc6 libgcc-s1 libpython3.10 libpython3.10-minimal libpython3.10-stdlib libstdc++6 python3.10 py
9 upgraded, 63 newly installed, 0 to remove and 120 not upgraded.
Need to get 82.7 MB of archives.
After this operation, 239 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

3. To install multiple version of Python, Update bashrc file

```
vi ~/.bashrc
```

```
ubuntu@ip-172-31-38-102:~$ vi ~/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
```

```
alias python3='/usr/bin/python3.7'
```

```
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias python3='/usr/bin/python3.7'
```

```
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 2
```

```
ubuntu@ip-172-31-38-102:~$ sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 2
update-alternatives: using /usr/bin/python3.7 to provide /usr/bin/python3 (python3) in auto mode
ubuntu@ip-172-31-38-102:~$
```

3. Copy api files from git and create gunicorn

i. Create the following folder.

```
mkdir aigoeye

mkdir aigoeye/apiImageProcessing

sudo chmod -R 777 aigoeye
```

```
ubuntu@ip-172-31-38-102:~$ mkdir aigoeye
ubuntu@ip-172-31-38-102:~$ mkdir aigoeye/apiImageProcessing
ubuntu@ip-172-31-38-102:~$ sudo chmod -R 777 aigoeye
```

ii. Copy the application code to above folder

iii. Install python venv

```
sudo apt install python3.7-venv
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo apt install python3.7-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3.7-distutils python3.7-lib2to3
The following NEW packages will be installed:
  python3.7-distutils python3.7-lib2to3 python3.7-venv
0 upgraded, 3 newly installed, 0 to remove and 120 not upgraded.
Need to get 2748 kB of archives.
After this operation, 3894 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

4. Create virtual environment

```
sudo apt install python-pip
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo apt install python-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpytho
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-pkg-resources python
Suggested packages:
  python-setuptools-doc python2-doc python-tk python2.7-doc binfmt-support
Recommended packages:
  python2-dev
The following packages will be REMOVED:
  python3-pip
The following NEW packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python-pip python-pkg-resou
0 upgraded, 10 newly installed, 1 to remove and 120 not upgraded.
Need to get 5488 kB of archives.
After this operation, 16.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
python3.7 -m venv apivenv
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ python3.7 -m venv apivenv
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$
```

```
source apivenv/bin/activate

pip install -r requirements.txt
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ source apivenv/bin/activate
(apivenv) ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ pip install -r requirements.txt
```

5. Deactivate virtual environment

```
deactivate
```

```
(apivenv) ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ deactivate
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$
```

6. Install mysql

```
sudo apt install mysql-server

sudo systemctl start mysql.service

sudo mysql
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-media
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-s
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-media
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-s
0 upgraded, 28 newly installed, 0 to remove and 120 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo systemctl start mysql.service
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
mysql>
```

7. create mysql users and passwords

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<password>';

ALTER USER 'root'@'localhost' IDENTIFIED WITH auth_socket;
```

```
mysql> CREATE USER 'aigoeye'@'%' IDENTIFIED BY 'Password@123';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'aigoeye'@'%'  WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> exit
Bye
```

```
CREATE USER '<username>'@'%' IDENTIFIED BY '<password>';

GRANT ALL PRIVILEGES ON *.* TO '<username>'@'%' WITH GRANT OPTION;

exit
```

```
mysql> CREATE USER 'aigoeye'@'%' IDENTIFIED BY 'Password@123';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'aigoeye'@'%'  WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
```

8. Install nginx

```
sudo apt update

sudo apt install nginx
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython3-
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter lib
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter lib
0 upgraded, 9 newly installed, 0 to remove and 120 not upgraded.
Need to get 697 kB of archives.
After this operation, 2395 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

9. Create aigoeye service from following location

```
cd    /etc/nginx/sites-available
sudo vi aigoeye
```

```
ubuntu@ip-172-31-38-102:~/aigoeye/apiImageProcessing$ cd /etc/nginx/sites-available
ubuntu@ip-172-31-38-102:/etc/nginx/sites-available$ sudo vi aigoeye
```

10. Update aigoeye file

11. Create link using following command

```
sudo ln -s /etc/nginx/sites-available/aigoeye /etc/nginx/sites-enabled
```

```
ubuntu@ip-172-31-38-102: sudo ln -s /etc/nginx/sites-available/aigoeye /etc/nginx/sites-enabled
ubuntu@ip-172-31-38-102:
```

12. Create aigoeye service and update file from aigoeye server

```
sudo vi /etc/systemd/system/aigoeye.service
```

```
ubuntu@ip-172-31-46-4:~$
ubuntu@ip-172-31-46-4:~$ sudo vi /etc/systemd/system/aigoeye.service
ubuntu@ip-172-31-46-4:~$
```

13. Update nginx.conf file

```
sudo vi /etc/nginx/nginx.conf
```

14. Start nginx, aigoeye service.

```
sudo ufw allow 4444
sudo ufw allow 3306
sudo ufw allow http
sudo ufw allow https
sudo nginx -t
```

```
ubuntu@ip-172-31-46-4:~$ sudo ufw allow 4444
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-46-4:~$ sudo ufw allow 3306
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-46-4:~$ sudo ufw allow http
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-46-4:~$ sudo ufw allow https
Rules updated
Rules updated (v6)
```

```
ubuntu@ip-172-31-36-84:/etc/systemd/system$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

```
sudo nginx -s reload
```

```
ubuntu@ip-172-31-36-84:/etc/systemd/system$ sudo nginx -s reload
```

```
sudo systemctl daemon-reload

sudo systemctl start nginx

sudo systemctl start mysql

sudo systemctl start aigoeye

sudo systemctl status aigoeye
```

```
ubuntu@ip-172-31-46-4:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-46-4:~$ sudo systemctl start nginx
ubuntu@ip-172-31-46-4:~$ sudo systemctl start mysql
ubuntu@ip-172-31-46-4:~$ sudo systemctl start aigoeye
```

```
[sudo] password for aigoeye:
● aigoeye.service - Gunicorn instance to serve aigoeye
     Loaded: loaded (/etc/systemd/system/aigoeye.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-09-20 18:10:44 CEST; 6 days ago
   Main PID: 2699423 (gunicorn)
      Tasks: 381 (limit: 308863)
     Memory: 3.4G
        CPU: 1w 5d 10h 20min 45.773s
     CGroup: /system.slice/aigoeye.service
             ├─2699423 /home/aigoeye/aigoeye/apiImageProcessingV4/apivenv/bin/python /home/aigoeye/aigoeye/apiImageProcessingV4/ap
             ├─2699425 /home/aigoeye/aigoeye/apiImageProcessingV4/apivenv/bin/python /home/aigoeye/aigoeye/apiImageProcessingV4/ap
             └─2699489 /home/aigoeye/aigoeye/apiImageProcessingV4/apivenv/bin/python /home/aigoeye/aigoeye/apiImageProcessingV4/ap

Sep 20 18:10:44 aigoeyeapi systemd[1]: Started Gunicorn instance to serve aigoeye.
Sep 20 18:10:44 aigoeyeapi gunicorn[2699423]: [2023-09-20 18:10:44 +0200] [2699423] [INFO] Starting gunicorn 21.2.0
Sep 20 18:10:44 aigoeyeapi gunicorn[2699423]: [2023-09-20 18:10:44 +0200] [2699423] [INFO] Listening at: unix:aigoeye.sock (269942
Sep 20 18:10:44 aigoeyeapi gunicorn[2699423]: [2023-09-20 18:10:44 +0200] [2699423] [INFO] Using worker: sync
Sep 20 18:10:44 aigoeyeapi gunicorn[2699425]: [2023-09-20 18:10:44 +0200] [2699425] [INFO] Booting worker with pid: 2699425
Sep 20 18:10:44 aigoeyeapi gunicorn[2699489]: [2023-09-20 18:10:44 +0200] [2699489] [INFO] Booting worker with pid: 2699489
Sep 20 18:10:45 aigoeyeapi gunicorn[2699425]: 2023-09-20 18:10:45.197641006 [W:onnxruntime:Default, onnxruntime_pybind_state.cc:54
Sep 20 18:10:45 aigoeyeapi gunicorn[2699489]: 2023-09-20 18:10:45.229776409 [W:onnxruntime:Default, onnxruntime_pybind_state.cc:54
lines 1-20/20 (END)
```

sudo systemctl status nginx

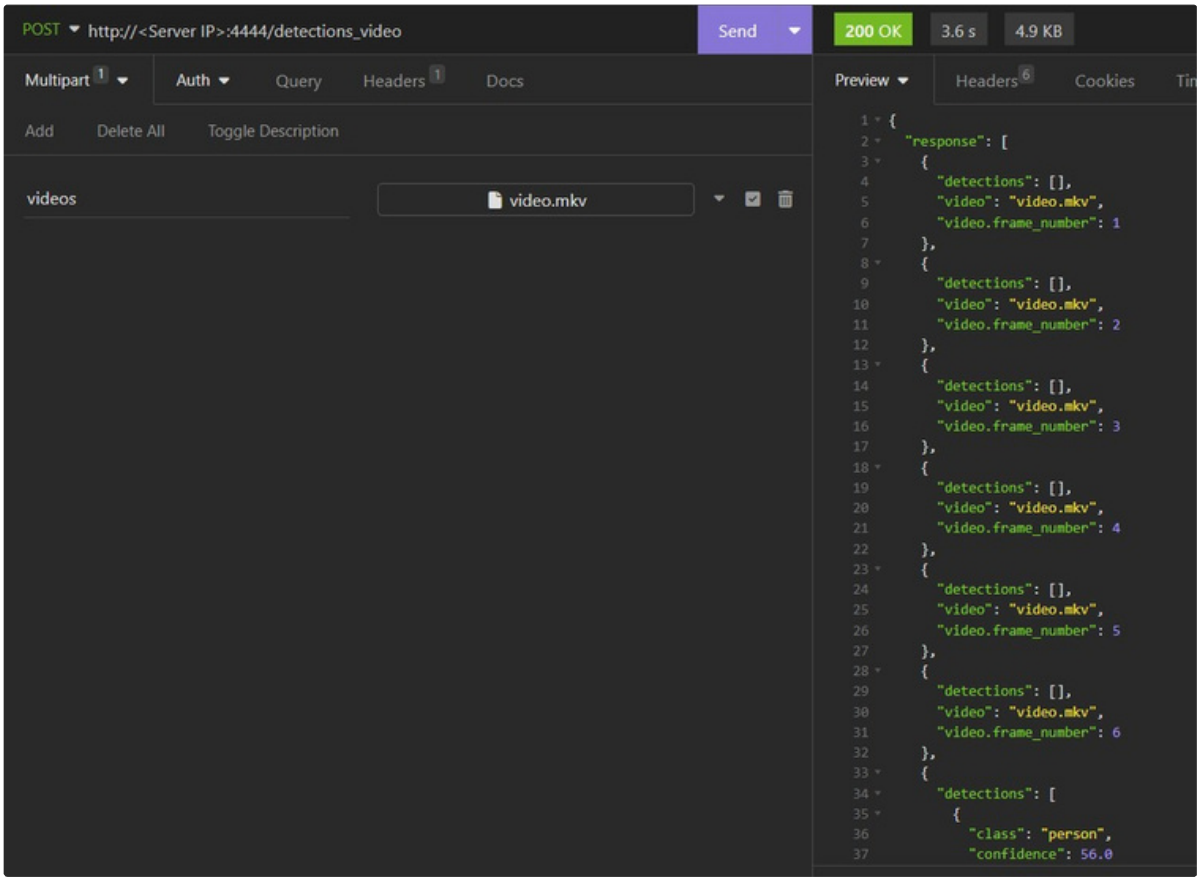
```
ubuntu@ip-172-31-46-4:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-10-04 09:31:48 UTC; 2min 11s ago
       Docs: man:nginx(8)
   Main PID: 3903 (nginx)
      Tasks: 5 (limit: 37951)
     Memory: 5.4M
        CPU: 30ms
     CGroup: /system.slice/nginx.service
             ├─3903 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─3905 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             ├─3906 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             ├─3907 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─3908 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Oct 04 09:31:48 ip-172-31-46-4 systemd[1]: Starting A high performance web server and a reverse
Oct 04 09:31:48 ip-172-31-46-4 systemd[1]: Started A high performance web server and a reverse
ubuntu@ip-172-31-46-4:~$
```

15. Update api url in plugin

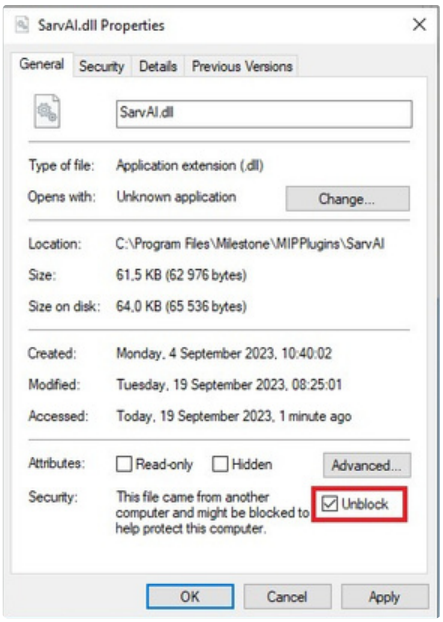
http://<servver IP>:4444/detections_video?=

16. To test API.



Plugin Installation

1. Unzip the SarvAI folder and copy it to <Milestone installation>\MipPlugins (usually c:\Program Files\Milestone\MipPlugins).
2. Right-click SarvAI.dll at <Milestone installation>\MipPlugins\SarvAI , click Properties and check the Unblock checkbox.



3. Restart the Event Server

Configuration

1. Open the Management Client.
2. Create alarm definitions that should generate the alarms (refer to Milestone documentaiton for how to create alarm definitions).
3. Go to MIP-plugins → SarvAI.

a. License Registration

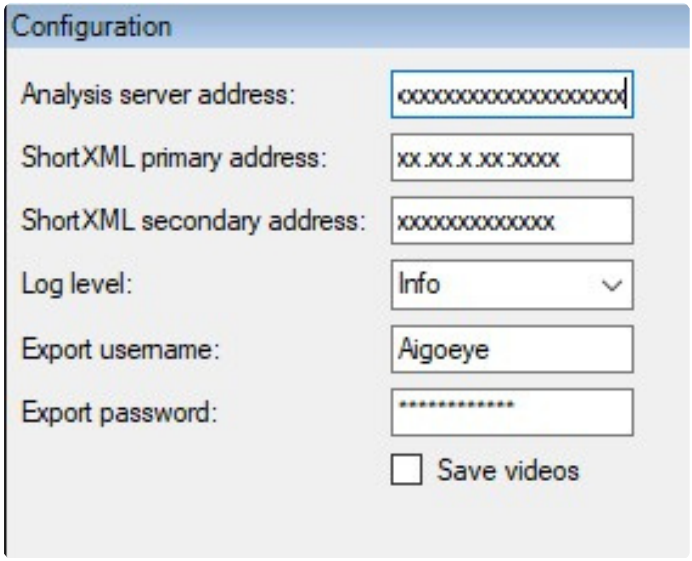
i. Enter the License key provided by the service provider and click Activate

b. Click on Settings

i. Enter the addresses of the analysis server that will be verifying the alarms and the ShortXML servers that will receive the verified alarms (example: 1.2.3.4:4444).

ii. Enter the username and password of either a Windows user or a Basic user, with permission to export videos from the cameras.

iii. Optionally select log level and whether to save exported videos on disk. Logs and saved videos can be found at c:\ProgramData\SarvAI. The logs are also included in the Event Server MIP log file.



c. Right-click on ShortXML definitions and select Add New...

d. Click on the newly created item and enter values for its properties.

i. *Object name*: any name that describes the source of the alarms.

ii. *Alarm definitions*: one or more alarm definition names separated by semicolon.

iii. *Send code:* Number with at most 8 digits identifying the sender.

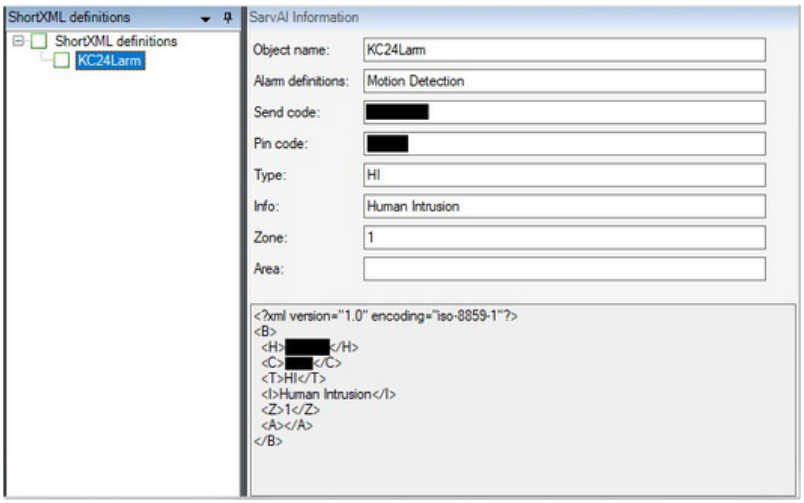
iv. *Pin code*: 4-character string that is used to validate the send code. Allowed characters are 0-9, A-Z.

v. *Type (optional):* 2-letter SIA code describing the type of alarms.

vi. *Info (optional)*: Any informative text that should be included in the ShortXML message, with at most 160 characters.

vii. *Zone (optional):* Number between 0 and 9999 identifying the zone.

viii. *Area (optional):* Number between 0 and 9999 identifying the area.



e. Optionally create more items with different parameters...