BOSCH
Invented for life

BVMS - ANPR by ISS

Author: Verhaeg Mario (BT-SC/PAS4-MKP)
Date: 26 August, 2019

1 Document information

Project	BVMS
Reference	n/a
Version	15
Last modified	26 August 2019

1.1 Version history

Version	Date	Who	Description
15	26 August 2019	Mario Verhaeg	Draft

2 Introduction

This guide assumes BVMS and ISS SecureOS are installed and functioning as separated systems. BVMS and ISS SecureOS can be installed on the same system.

2.1 Functionality

SecurOS sends events from analytics to BVMS, which can be translated into alarms and/or stored into the logbook.

2.2 Events received

Event	Included metadata	Description
License Plate Detected	License plate, Direction, DirectionID, SourceID	General License plate event
License Plate on watchlist detected	License plate, Direction, DirectionID, SourceID	Watchlist License plate event

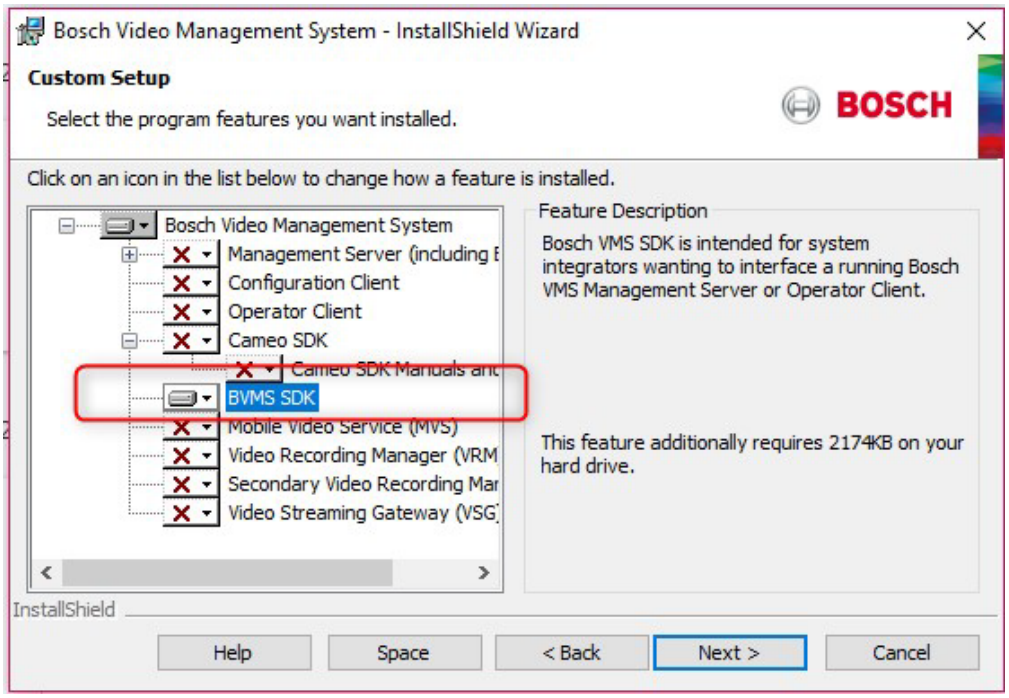
In addition to SecureOS Auto BVMS can also receive events and metadata from SecureOS Transit, SecureOS Cargo and SecureOS FaceX.

3 Installation

3.1 ISS SecurOS

3.1.1 Install the BVMS SDK

The BVMS SDK needs to be installed on the ISS SecureOS system, if it is not running on the same server as the BVMS Management Server. The BVMS SDK can be installed as part of the BVMS installation package.



The integration between ISS SecureOS and BVMS is based on Javascript technology. To enable the use of the Javascript plugin a DLL file needs to be registration.

1. Copy Bosch.Vms.VirtualInputsServer.dll from the BVMS Management Server C:\Program Files\Bosch\VMS\bin() to the ISS SecureOS server (C:\Program Files\Bosch\VMS\bin)
2. Register the DLL by calling from the root of the .NET framework directory:

```
regasm.exe /codebase “C:\Program Files\Bosch\VMS\bin\Bosch.Vms.VirtualInputsServer.dll“
```

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319>regasm.exe /codebase "C:\Program Files\Bosch\VMS\bin\Bosch.Vms.VirtualInputsServer.dll"
Microsoft .NET Framework Assembly Registration Utility version 4.7.3190.0
for Microsoft .NET Framework version 4.7.3190.0
Copyright (C) Microsoft Corporation.  All rights reserved.

Types registered successfully

C:\Windows\Microsoft.NET\Framework\v4.0.30319>
```

3.1.2 Sending events

Open the ISS configuration from the Windows task bar.

Browse to the VB/JScript programs node.

Sending events for all detected license plates and trigger alarm on watchlist events

Create a new JScript program and use the code-example below. Adjust the necessary parameters, for example the IP address, username and password of the BVMS server.

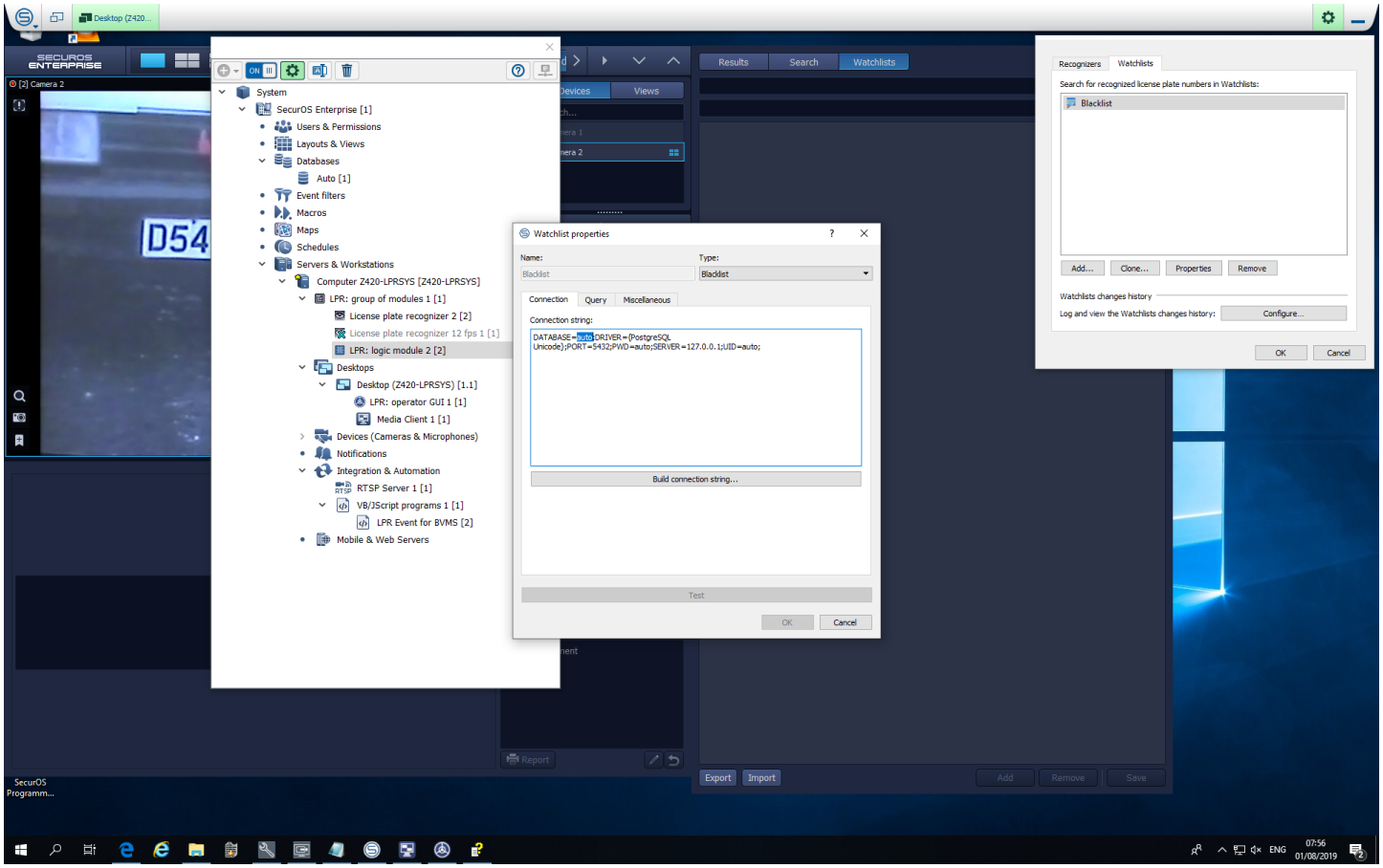
```
var BVMS_server = "192.168.178.81:5390"; // IP-address and port of BVMS
var BVMS_login = "Admin"; // BVMS username
var BVMS_password = "xyz"; // BVMS username password

var pass
function Init()
{
    // Execute function LPPass when a license plate is detected for all cameras
    Core.RegisterEventHandler("LPR_CAM","*","CAR_LP_RECOGNIZED","Detection");
    // Execute function Watchlist when a license plate on a watchlist is detected
    Core.RegisterEventHandler("LPR_LOGIC","*","CAR_LP_FOUND","Watchlist");
}
function Detection(event)
{
    // Store license plate event in global variable pass
    pass = event

    // Send data to BVMS Logbook
    var VirtInput;
    // BVMS VirtualInputID for Logbook is equal to the ISS SecureOS Camera ID
    var inputNr = event.sourceid;
    var alarmId = 1;
    var dataString1 = "";
    var dataString2 = event.number;          // License plate
    var dataString3 = event.direction_id;    // Direction ID
    var dataString4 = event.direction_name; // Direction Name
    var dataString5 = event.sourceid;        // ISS SecureOS Camera Source ID
    var dataString6 = ""; // reserved
    var dataString7 = ""; // reserved
    var dataString8 = ""; // reserved
    var dataString9 = ""; // reserved
    var dataString10 = ""; // reserved

    Log.Debug("Detection", inputNr, dataString1, dataString2, dataString3, dataString4,
dataString5, dataString6, dataString7, dataString8, dataString9, dataString10, alarmId);
    VirtInput = new ActiveXObject("Bosch.Vms.VirtualInputsServer");
    VirtInput.Connect(BVMS_server, BVMS_login, BVMS_password);
    VirtInput.SendData(inputNr, dataString1, dataString2, dataString3, dataString4,
dataString5, dataString6, dataString7, dataString8, dataString9, dataString10, alarmId);
}
function Watchlist(event)
{
    var VirtInput;
    // BVMS VirtualInputID for Logbook is equal to the ISS SecureOS Camera ID + 1000
    var inputNr = (parseInt(event.sourceid) + 1000).toString();
    var alarmId = 1;
    var dataString1 = "";
    var dataString2 = event.number;          // License plate
    var dataString3 = event.direction_id;    // Direction ID
    var dataString4 = event.direction_name; // Direction name
    var dataString5 = event.sourceid;        // ISS SecureOS Camera Source ID
    var dataString6 = event.database_type;  // Watchlist type
    var dataString7 = event.database_name;  // Watchlist name
    var dataString8 = ""; // reserved
    var dataString9 = ""; // reserved
    var dataString10 = ""; // reserved
```

```
    Log.Debug("Watchlist", inputNr, dataString1, dataString2, dataString3, dataString4,
dataString5, dataString6, dataString7, dataString8, dataString9, dataString10, alarmId);
    VirtInput = new ActiveXObject("Bosch.Vms.VirtualInputsServer");
    VirtInput.Connect(BVMS_server, BVMS_login, BVMS_password);
    VirtInput.SendData(inputNr, dataString1, dataString2, dataString3, dataString4,
dataString5, dataString6, dataString7, dataString8, dataString9, dataString10, alarmId);
}
```

The script is based on one single Watchlist, but can be extended to multiple Watchlists as well.

3.1.3 Configuring ISS SecureOS Watchlists

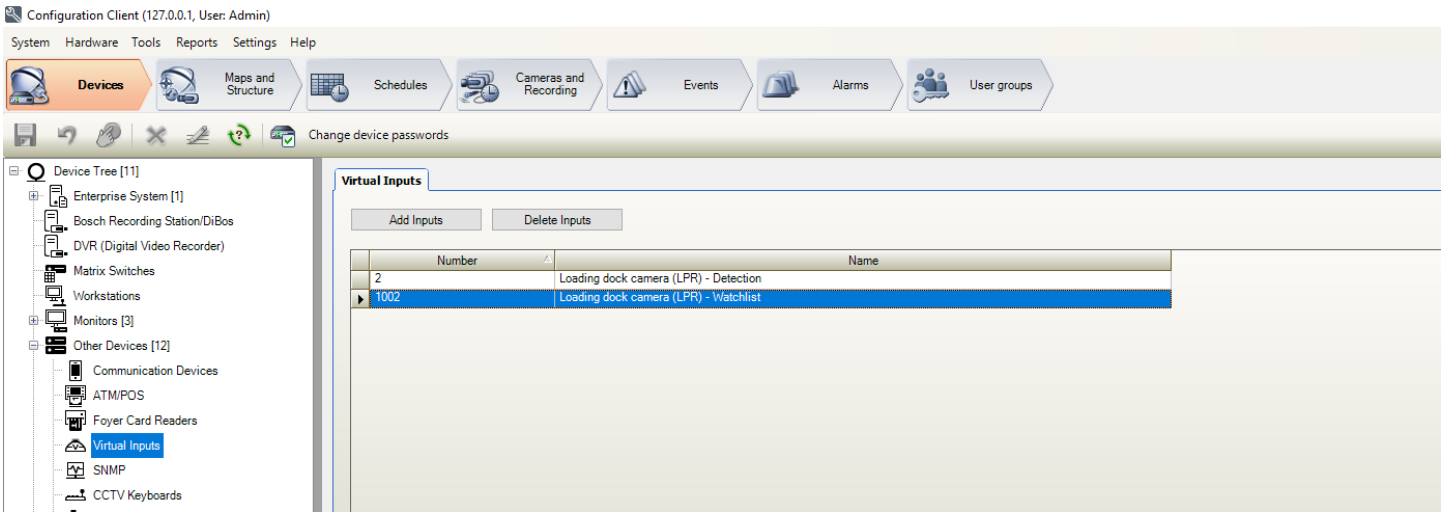
The Watchlists in ISS SercureOS are not enabled by default. The ISS SecureOS Auto user guide describes how to enable the Watchlists (section 4.1.5.2 Watchlists and 5.1.5.2.1 External Database Connection Example).



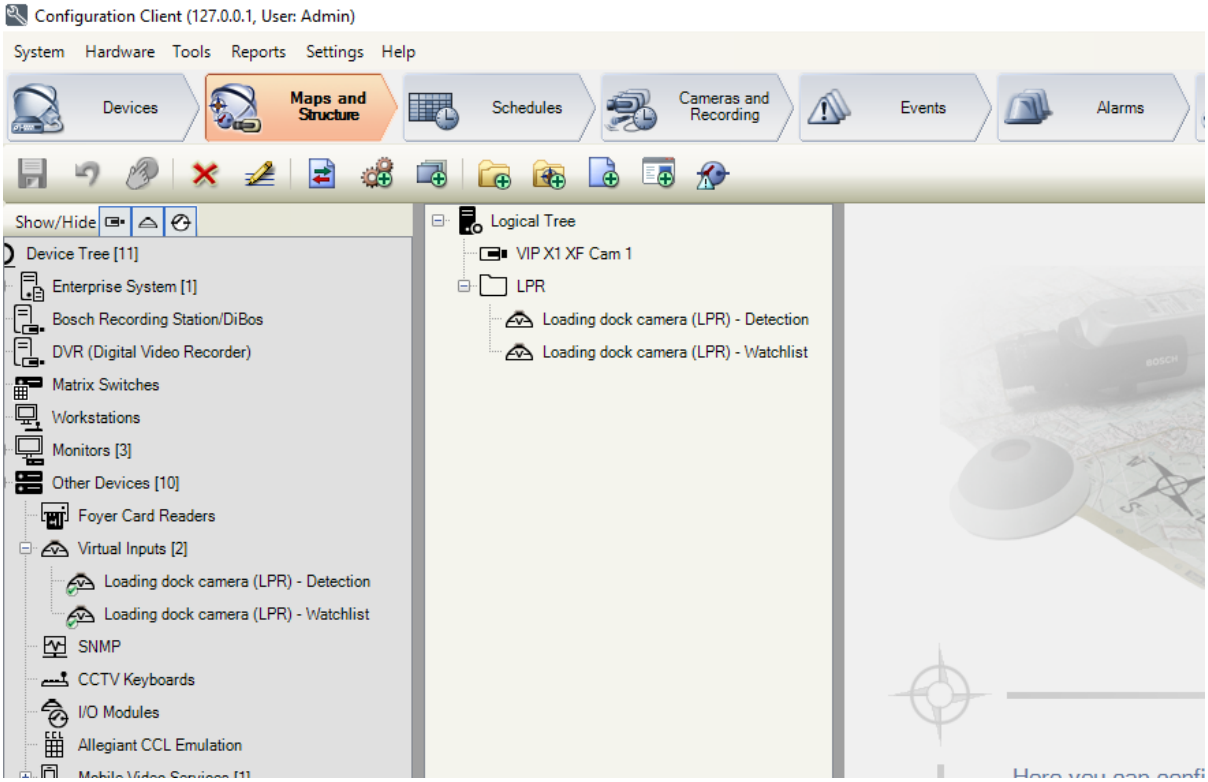
3.2 BVMS - Configuration

3.2.1 Creating virtual inputs

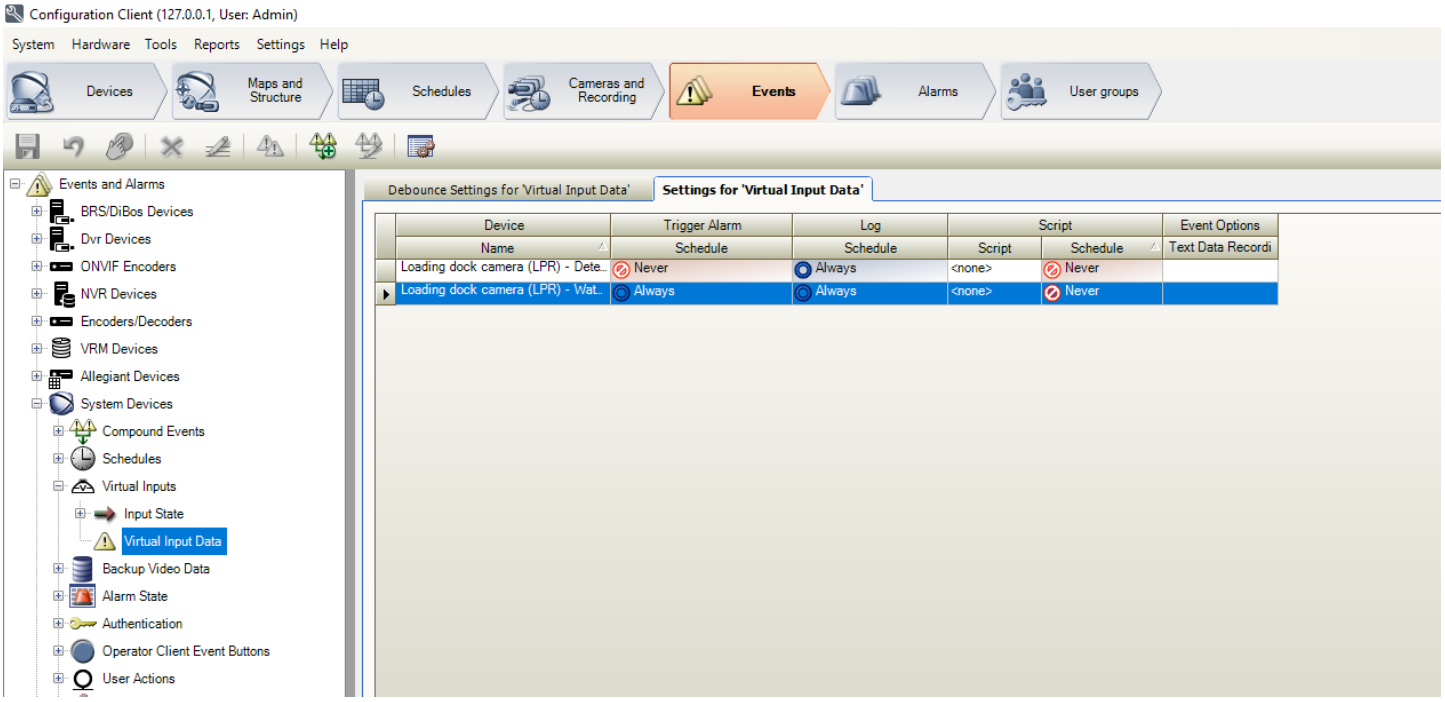
Create the necessary virtual inputs in the BVMS device tree. In the example below two virtual inputs are used and named after two cameras for which ISS SecurOS is providing ANPR.



Add the virtual inputs to the logical tree. It is recommended to add, per ANPR camera, one virtual input which receives all ANPR events and one virtual input which only receives and event when a watchlist is triggered.



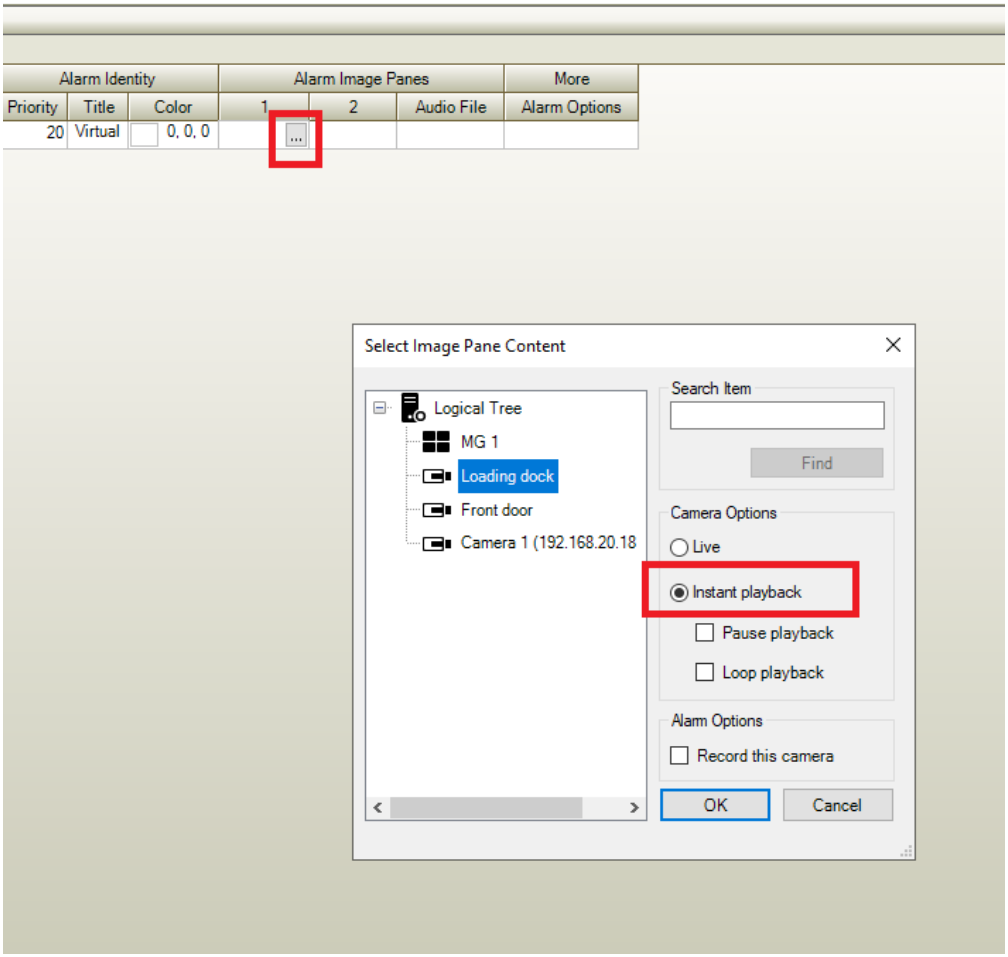
Trigger alarms (if necessary) for the related virtual inputs. The virtual input used for receiving all ANPR events will not be configured to raise an alarm. The virtual input which receives the watchlist events will be configured to raise an alarm.



3.2.2 Creating alarms

Configure the appropriate alarm settings (alarm settings are described in the BVMS Configuration Manual).

Instant-playback can be used to allow the operator to immediately verify the incident as it is happening.



4 Operation

4.1 Detecting incidents

When the watchlist is configured the BVMS alarms are visualized using the BVMS alarm management mechanism. The license plate itself is showed and stored as Text Data.

4.2 Investigation

The logbook search or search video by event mechanism can be used to search for a specific license plate. License plates are stored as "Data 2". The picture below shows an example of the search parameters.