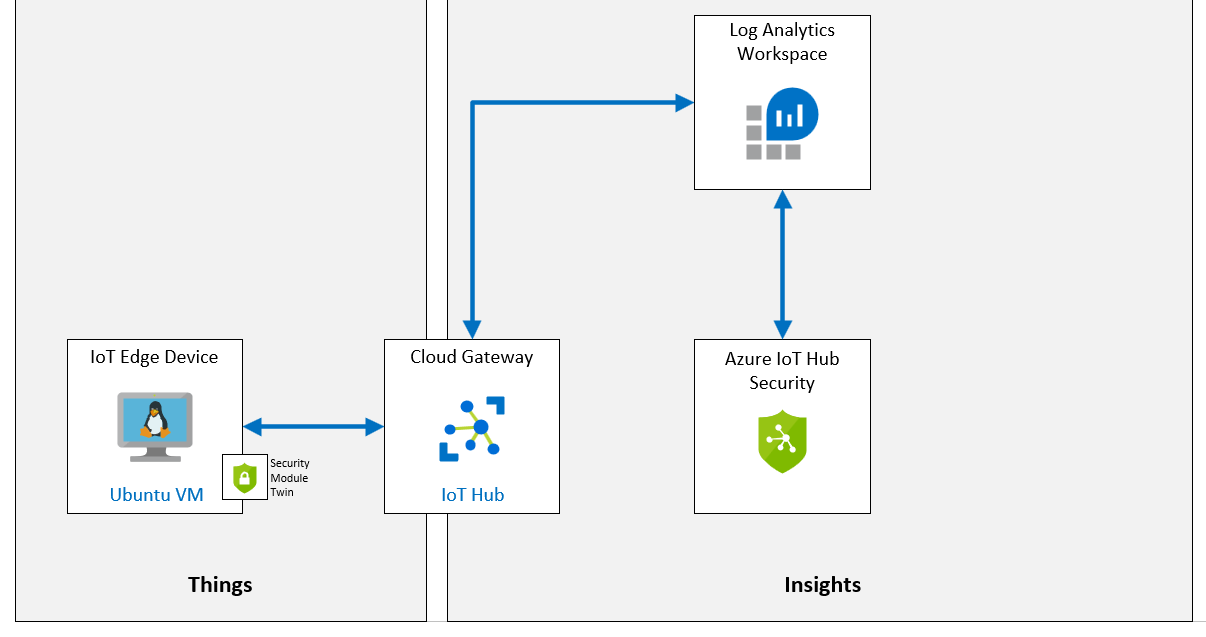
Detect Device Tampering with Azure Defender for IoT



1. **create an IoT Hub: Done**
2. **Enable Azure Defender** for Io: Done
3. Create and register a new Device: Done
4. Create a **Security Module Twin: Done**
5. Install C#-based **Security Agent** on a Linux Device: Done
6. Configure **monitored resources: Done**
7. Create custom **alerts: Done**
8. Create **a console app** to trigger the alert: Done
9. **Review the alert** in the Azure Defender for IoT: Done
10. Create an IoT Hub: Done

Rg-az220

Iot-az220-testing-0001

#### **Enable Azure Defender for IoT: Done**

click **iot-az220-training-{your-id}**.

under **Security**, and then click **Overview**.

If the **Secure your IoT solution** button is displayed, click **Secure your IoT solution**, and then refresh your browser window when prompted.

After a few moments you will see the message **Onboarding succeeded for this IoT hub, please refresh for changes to take effect**

#### **Task 2: Log Analytics creation: Done**

When Azure Defender for IoT is turned on, an **Azure Log Analytics workspace** should be created to **store raw security events, alerts,** and **recommendations** for your IoT devices, IoT Edge, and IoT Hub.

under **Security**, click **Settings**..

 click **Data Collection**.

1. In the **Workspace configuration** section, under **Choose the Log Analytics workspace you wish to connect to:**, ensure the toggle button is set to **On**.
2. **Workspace** dropdown, click **Create New Workspace**.
3. **Pricing tier**, ensure **Pay-as-you-go** is selected
4. **Access to raw security data** is checked.
5. **In-depth security recommendations and custom alert** is checked.
6. **IP data collection** is checked.
7. click **Save**

### **Exercise 3: Create and Register a New Device: Done**

you will be setting up a virtual machine that you will then **use to simulate an IoT device**

you will use this device to **measure vibrations on a conveyor belt**.

**rg-az220vm**

**vm-az220-testing-edge0002-0001**

**Ubuntu Server 18.04 LTS - Gen1** image.

#### **Task 2: Register New Devices**

**IoT devices** pane, click **+ New**

**Device ID**, enter **vm-az220-testing-edge0002-0001**

**Symmetric key** for authentication

**Save**

### **Exercise 4: Create a Security Module Twin: Done**

 The security module twin holds all the information relevant to device security for each of your devices

The security module twin (**azureiotsecurity**) can be created:

* Use the [Module batch script](https://github.com/Azure/Azure-IoT-Security/tree/master/security_module_twin).
* **Manually**

**IoT devices**

**+ Add Module Identity**.

**Module Identity Name**, enter **azureiotsecurity**

**Get Primary key of DeviceID**

**Get Hostname of Iot Hub**

### **Exercise 5: Deploy Azure Defender for IoT C# Security Agent: Done**

Security agents support the following features:

* **Collect raw security events** from the underlying **Operating System** (Linux, Windows). To read more about available security data collectors, see Azure Defender for IoT agent configuration.
* **Aggregate raw security events into messages sent through IoT Hub**.
* **Authenticate with existing device identity**, or a dedicated module identity. To read more, see Security agent authentication methods.
* **Configure remotely** through use of the **azureiotsecurity** module twin. To read more, see Configure an Azure Defender for IoT agent.

#### **Logging into IoT Device - Linux VM**

Ssh [adminuser@20.102.57.83](mailto:adminuser@20.102.57.83)

#### **Add Symmetric Keys to your device**

echo "<primary\_key>" > s.key

 download the most recent version of Security Agent for C# to your device

wget https://github.com/Azure/Azure-IoT-Security-Agent-CS/releases/download/0.0.6/ubuntu-18.04-x64.tar.gz

extract the contents of the package and navigate to the /Install

tar -xzvf ubuntu-18.04-x64.tar.gz && cd Install

add execute permissions to the InstallSecurityAgent script

chmod +x InstallSecurityAgent.sh

This script performs the following function:

* Installs prerequisites.
* Adds a service user (with interactive sign in disabled).
* Installs the agent as a Daemon - assumes the device uses **systemd** for service management.
* Configures **sudo users** to allow the agent to do certain tasks as root.
* Configures the agent with the authentication parameters provided.

## sudo ./InstallSecurityAgent.sh -i -aui Device -aum SymmetricKey -f ../s.key -hn iot-az220-testing-00001.azure-devices.net -di vm-az220-testing-edge0002-0001

reconnect to Device and check status of security module:

systemctl status ASCIoTAgent.service

Check the status connection on the Iot Device

### **Exercise 6: Configure Solution Management: Done**

Azure Defender for IoT provides end-to-end security for Azure-based IoT solutions.

Once enabled on your IoT Hub, **Azure Defender for IoT** **automatically identifies other Azure services,** also connected to your IoT Hub and related to your IoT solution.

#### **IoT Hub**

**Security**, click **Settings**.

click **Monitored Resources**.

 click **Edit**.

**Solution Management**

**Subscriptions, Resource groups, click rg-az220vm.**

click **Apply**.

#### **Task 2: View Azure Defender for IoT in Action:**

under **Security**, click **Overview**

**Security**, to view the monitored resources, click **Settings** and then click **Monitored Resources**

**Edit**

**Add more resource**

**Check Security> Overview**

### **Exercise 7: Introduce custom alerts: Done**

#### **Why use custom alerts?**

Azure Defender for IoT allows you to translate this understanding into a device behavior policy and alert on any deviation from expected, normal behavior.

you will create a custom alert that monitors **Device to Cloud** messages sent via the **MQTT** protocol.

IoT Hub

**Security**, click **Settings**.

click **Custom Alerts**.

Security groups enable you to define logical groups of devices, and manage their security state in a centralized way

To add a custom alert to the default security group, click **default**.

click **Create custom alert rule**

click **Number of device to cloud messages (MQTT protocol) is not in allowed range**.

Under **Required Properties**, in the **Minimal Threshold** field, enter **1**.

Under **Maximal Threshold**, enter **5**.

In the **Time Window Size** dropdown, click **00:05:00**.

At the bottom of the **Create custom alert rule** pane, click **OK**.

click **Save**

#### **Task 1: Register New IoT Device: Done**

On the **Create a device** blade, under **Device ID**, enter **sensor-th-0070**

**Device Twin**

To add the device to the **default** security group, insert the following JSON between the **version** and **properties** fields:

"tags": {

"SecurityGroup": "default"

},

#### **Task 2: Configure a Cave Device App**

**CaveDevice**

private readonly static string deviceConnectionString = "<your device connection string>";

dotnet run

### **Exercise 9: Review Azure Defender for IoT Alerts**

### **Iot Hub**

**Security**, click **Overview**.