AIR QUALITY MONITORING

USING IOT

ABISHEK.T

au812921106002

abishekmusiri203@gmail.com

INTRODUCTION:

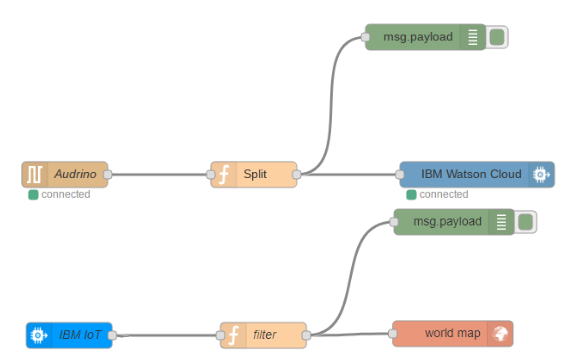
* As the name suggests, embedded describes something that is connected to another item. An embedded system is a piece of computer hardware that also contains software. A stand-alone unit or a component of a bigger system can both be an embedded system. An embedded system with a microcontroller or microprocessor is designed to carry out a certain purpose. For instance, a fire alarm is an integrated device that only detects smoke. An embedded system is similar to a computer system in that it is primarily designed to accomplish specific functions such as controlling data in various electronics-based systems, accessing data, processing, and storing data. Embedded systems are hardware and software combinations that are designed to perform a certain set of functions. The embedded system's most essential feature is that it regenerates the output in a very short amount of time. Many embedded systems will be encountered in our daily lives.OCK DIAGRAM:

Node-RED Vulnerabilities:

Node-RED is “a programming tool for wiring together hardware devices, APIs and online services”, which provides a way of “low- code programming for event- driven applications” [36]. As an open-source platform, Node-RED is mainly tar- geted for deployment as a single-user platform, although it is also available on the IBM Cloud platform [23].

Node-RED platform:

A node is a reactive Node.js application triggered by receiving messages on at most one input port (dubbed source ) and sending the results of (side-effectful) computations on output ports (dubbedsinks ), which can be potentially multiple, unlike the input port. Figure 3 illustrates the code structure of a Node-REDnode. A special type of node without sources and sinks, called configuration node, is used for sharing configuration data, such as login credentials, between multiple nodes. A flow is a representation of nodes connected together. End users can either create their own flows on the platform’s environment or deploy existing flows pro- vided by the official Node-RED catalog [33] and by third parties. In Node-RED, contexts provide a shared communication channel between different nodes without using the explicit messages that pass through a flow [40]. Therefore the node wiring visible in the user interface reflects only a part of the information flows that are possible in the flow. It introduces an implicit channel that is not visible to the user via the graphical interface of a flow. Node-RED defines three scope levels for the contexts

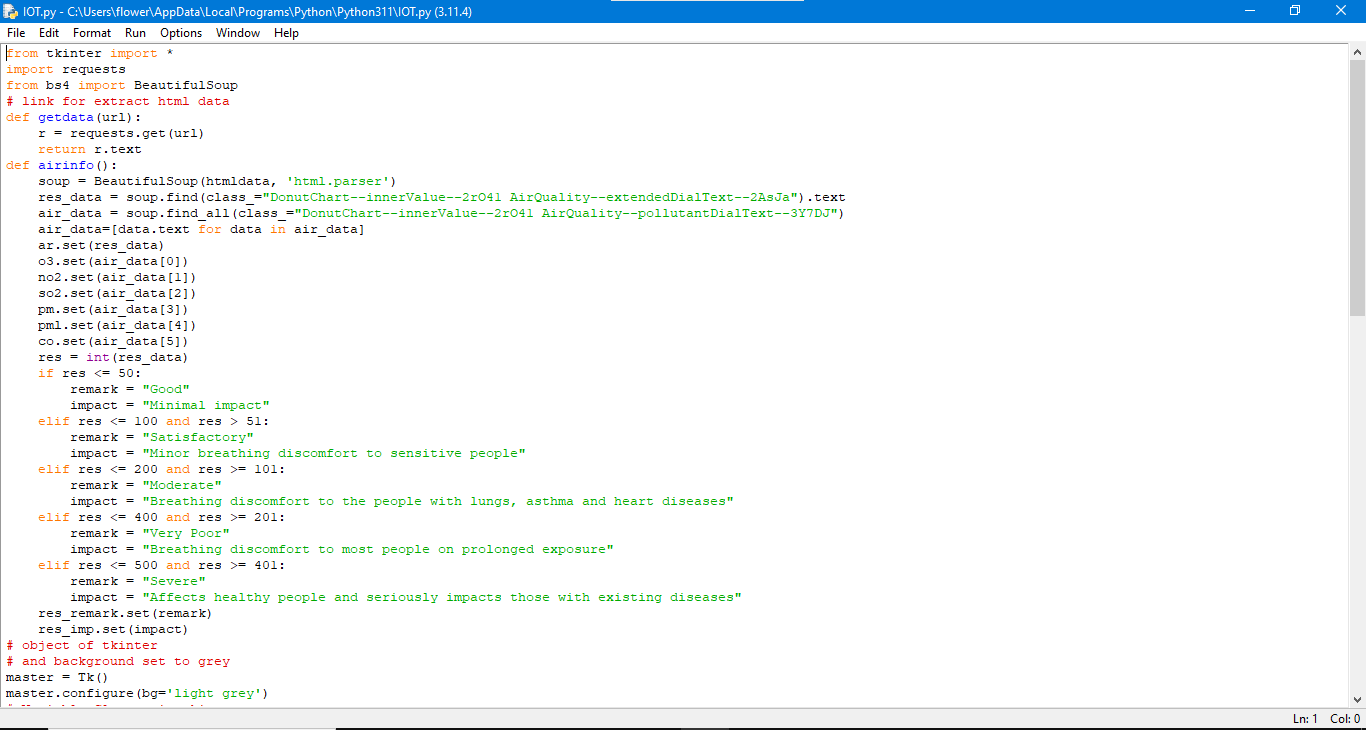


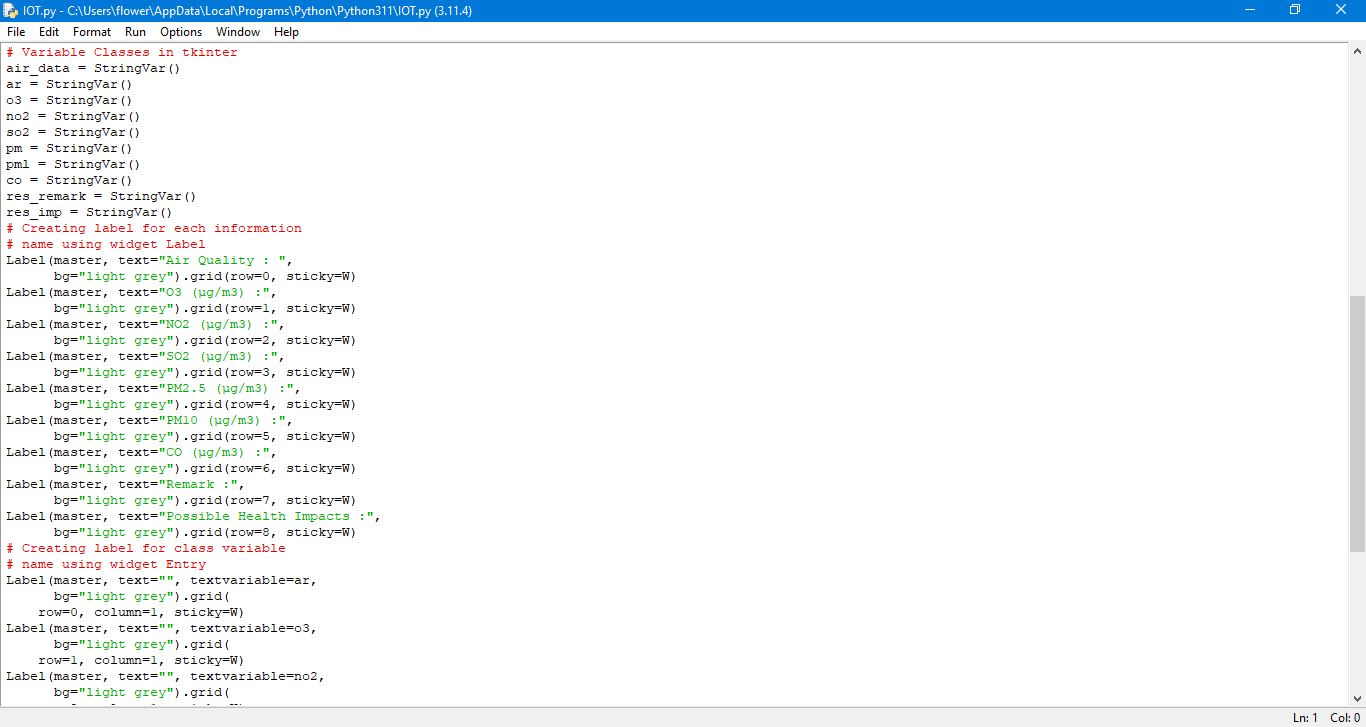
The provided policies can later be vetted by the platformand the user, before deploying the node. SandTrap [3] offers a pol- icy generation mechanism to aid developers in designing the policies, enabling both baseline and advanced policies customized by developers or users to express fine-grained app- specific security goals. In the following, we discuss Node-RED attacks and vulnerabilities that mo- tivate enriching the policy mechanismwith different granularity levels. These policies will further be formalized in Section 3.

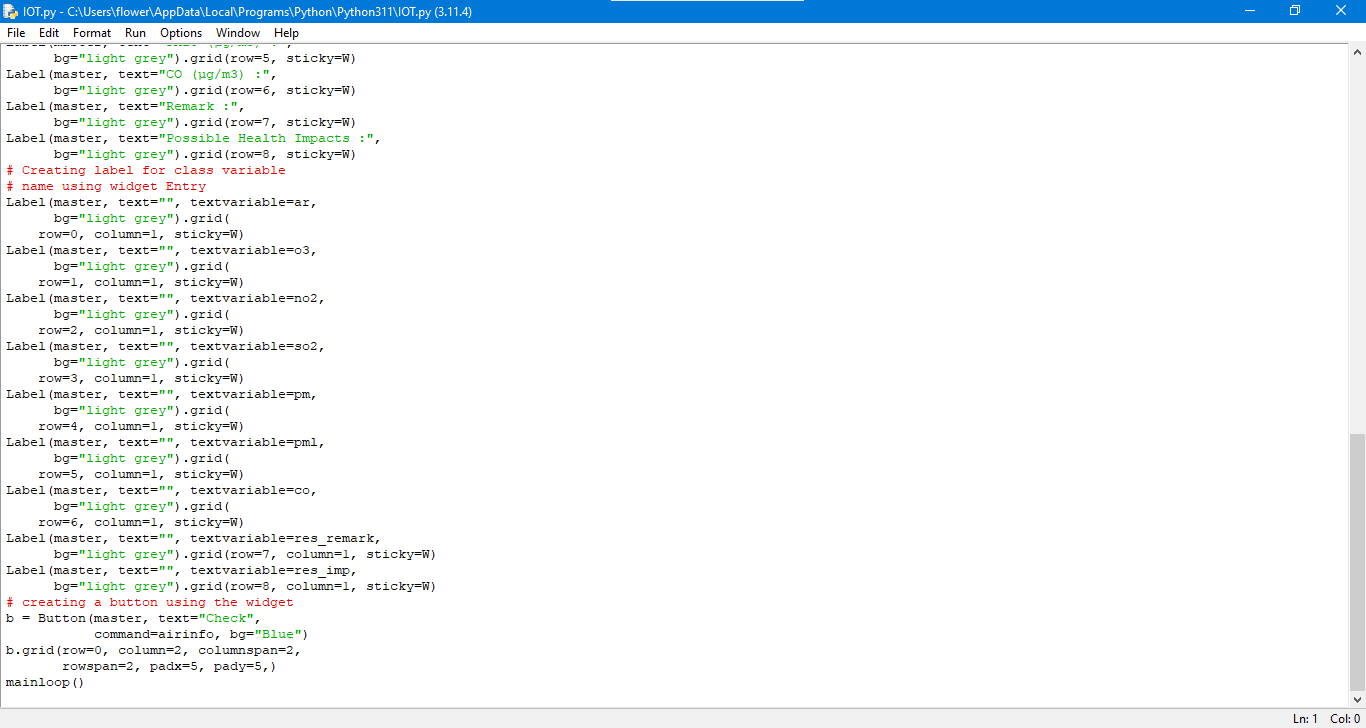
Platform-level isolation vulnerabilities:

All APIs provided by the underlying runtimes, Node-RED and ⟨ ⟩ Node.js, are accessible for node developers, as well as the incom- ing messages within a flow. As shown in Figure 6a, there are various attack sce- narios for malicious nodes [3]. At the Node.js level, an attacker can create a ma- licious Node-RED node including.

Program:







OUTPUT:

