Graphical Databases for IoT Device Management and Anomoly Detection

Colter Snyder
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
csnyder1@mines.edu

***Abstract*—The wide spread proliferation of Internet of Things (IoT) devices have brought up many questions and concerns about their security and how much data they truly collect. Many papers have been writen to detect IoT devices and catalog them, however none use a graphical database in order to do so. Using a graphical database provides many benefits over current systems and allows for new questions to be answered.**

***Index Terms*—IoT, graphical databases, databases, systems**

I. Introduction

Smart homes are all the rage now adays. There are smart devices everywhere, from lights to door locks. It seems that these devices, which are refered to by their collective as Internet of Things or IoT devices, are everywhere. With all these new devices come a whole slew of concerns about privacy and security [**IoTInspector**]. There are many papers that explore these concerns, what this paper seeks to explore is how those findings are aggregated and stored. There has been a lot of success at using graphical databases due to their speed for querying on certain questions regarding big data.

II. Related Works

There are many current systems that implement different components of the general idea of graphical databases for IoT device management andanomoly detection, but none put these components together. *IoT Inspector* is a great tool that performs the task of monitoring using a form of a relational database, but not a graphical one [**IoTInspector**]. The authors of *The Graph of Things* created such a system for aggregating IoT devices worldwide [**GraphofThings**]; However, these systems are not built for small networks or home users.

III. Methodology

The first step of the methodology was to determine the technology stack. It was determined that NMap would be used to scan for devices, TCPDump would be used for packet capture, Python for programming, and Neo4J for the database technology. The database technology was the most difficult to determine due to the unique challenges faced when using a graphical database. The first idea was to use the Resource Description Framework (RDF) as the base. There currently does exist an ontology using RDF for IoT devices [**RDFIoT**]. Indeed, using an ontology for IoT devices is very useful for cataloging at scale [**IoTSmartOntology**]. However, this project is not designed for scale, but for the userspace. For this RDF has some drawbacks such as needing a separate node and edge for details about a parent node. This is not optimal for both performance and size. It was determined, then, to go with a graphical model that allows for information to be included with each node. This is where Neo4J comes in. With attributes being stored in JSON files connected to the nodes makes it optimal for this use.

IV. Results

V. Analysis

VI. Future Work

VII. Conclusion