Graph Databases for Use with Timeseries IoT Datasets

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 what they do with data. Many solutions have appeared using various solutions such as IoT Inspector [1]. However, these solutions don't make use of the unique capabilities and efficiencies that graph databases provide. This paper seeks to use a graph database in order to answer challenging questions about IoT devices particularly within the realm of timeseries datasets.

Index Terms—IoT, graphical databases, databases, systems

I. INTRODUCTION

Smart homes are everywhere now adays, from lights to door locks, TVs to speakers. It seems that these devices, which are refered to by their collective as Internet of Things or IoT devices, are in every home. With all these new devices come a whole slew of concerns about privacy and security [1]. There are many papers that explore these concerns, what this paper seeks to persue is how to efficiently analyze data collected from these devices such that people may infer various aspects about their devices. In particular, this paper seeks to see what what info can be garnered from timeseries datasets. Such aspects could include anything from what the device is doing to answering if a device is attacking the network and how.

It was decided that using a graph database could provide answers to these questions more efficiently and easily than a relational database. The primary advantage for use in this paper is the fact that graph databases are great for use with densely connected data [2]. On top of this, they are very quick to query and will give a result relatively quickly compared to other solutions [2]. In particular, the use for home users and smaller networks was examined by asking what queries could be run on a smaller dataset as well as what would be reasonable for a home user as far as time and efficiency.

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 [1]. The authors of *The Graph of Things* created such a system for aggregating IoT

devices worldwide [4]; However, these systems are not built for small networks or home users. What this paper seeks to explore is what questions can a graph database answer for these smaller networks, home users, and generally users with smaller amounts of IP addresses being used. The queries and dataset going forward were chosen with these thoughts in mind.

III. METHODOLOGY

This projects main goal was to see the about the ease of use for a graph database for answering certain questions. It was hypothesied that fairly simple queries to the database would allow the author to gain large amounts of information to be analyzed and from which many conclusions could be drawn.

The first towards determining this hypothesis was choosing the technology that was to be used. It was decided to use Neo4J and Python for the tech stack. Neo4J was chosen due to its unique feature of having practically unlimited properties attached to both the nodes and the edges. This was particularly useful when working with 5-tuples where the nodes could hold the addresses and the edged could hold the ports, protocals, and a timestamp. Python was chosen due to its ease of use, quick prototyping, and library support for Neo4J.

The next step was aquaring the data. In this regard, after a long time of testing and consideration, it was determined that using a prebuilt dataset would be the best and most time efficient. A perfect dataset was found with BoT IoT [3]. This dataset includes both simulated and real world IoT devices as well as showing many attacks on these devices. Using such a dataset would allow for getting great amounts of patterns for which queries could be written to detect. In addition, being timeseries based, it was perfect for this paper.

With both the tech stack and dataset decided, the next step was to decide what types of queries should be written to get the most out of the potentials of both the dataset and graph databases as a whole. Some basic queries were written with some queries left to future work due to lack of knowledge and time on the part of the author. The basic queries included queries such as flows to and from an IP address to determine whether a device was a sink, source, or equal with regards to flow direction, how many flows were sent/received per second, how many ports were contacted, and the number of incoming flows to a particular address. These queries were chosen in

particular to look at two certain attacks: DDoS and Network Scan attacks. If an address were a sink, it could be a target, if it was a source it could be an attacker for DDoS. However to truly determine this flows per second and incoming flows would also need to be detected. For Network Scans, if the number of ports an IP address sends a flow to is abnormally high, it is a likely indicator that a network scan is being performed.

The last step was writting the program and aquiring the results. This produced some very interesting conclusions and was satisfying to the author. This also happened to be the longest part of the methodology as parsing all the data took a relatively long time, over a day and a half to parse 1,000,000 flows into the database. Once parsed, however, the queries were very quick with the longest taking only a matter of seconds to execute.

IV. RESULTS

With the experiments done and the data aquired there were some interesing results that were discovered. One such discovery was that, within the dataset being used, there were clear sinks and sources for flows as shown in figures 1 and 2.
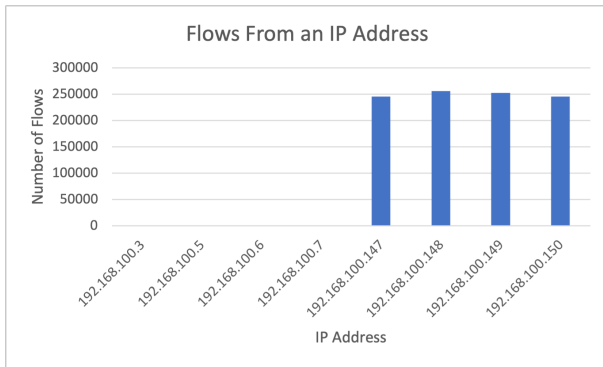

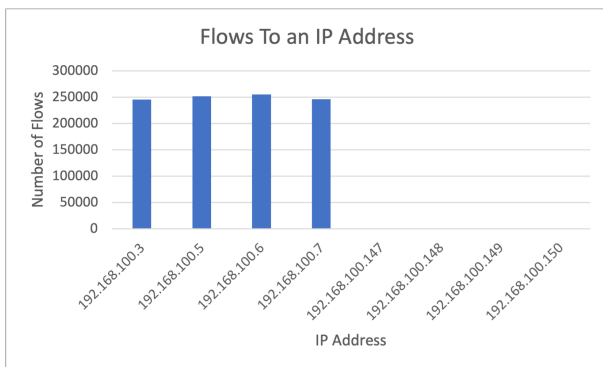Fig. 1. The number of flows that were sent from particular IP addresses

Fig. 2. The number of flows that were sent to particular IP addresses

A clear pattern of sending and receiving was being established in the data which is a potential indicator of some sort of attack whether DDoS, Network Scanning or otherwise. To

determine this further, the average number of flows per second was determined as shown in figures 3 and 4.

While it is good to see how much data was being sent overall, it doesn't really tell very much about what's happening. In order to determine something like a DDoS attack, the average packets sent pers second was determined to be more useful.
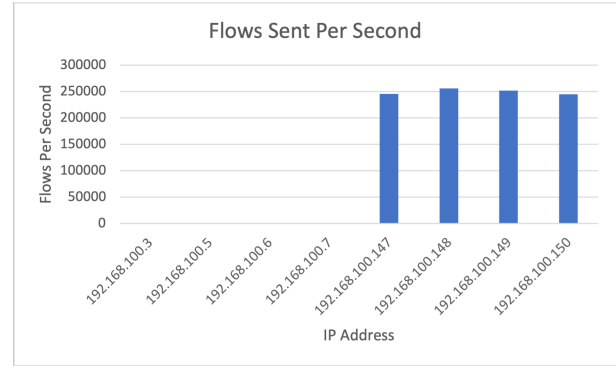

Fig. 3. The number of flows that were sent from particular IP addresses per second
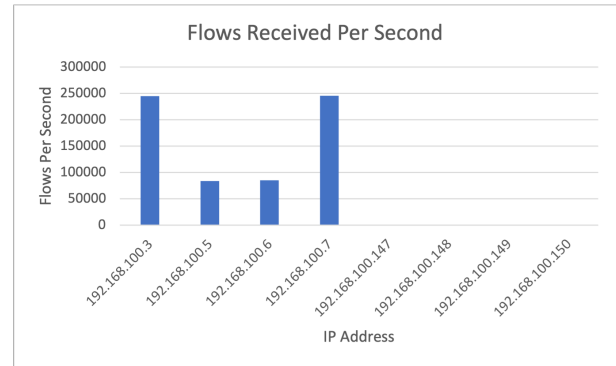

Fig. 4. The number of flows that were sent to particular IP addresses per second

As with figures 1 and 2, these figures also showed clear trends with definitive sinks and sources, however this was not enough to fully determine what was going on with this data. To make a more definitive conclusion the number of ports each IP address contacted was found as shown in figure 5.

This figure provided a more definitive idea about what was going on for some of the IP Addresses. In particular, 192.168.100.148 contacted the most ports likely indicating that it was performing a network scan on one of the other networks. Similarly could be said for 192.168.100.149 and 192.168.100.150 with less confidence.

With the question of network scans out of the way, the last question remained. Was it possible that any of the sinks were being DDoSed? For a DDoS to occur there should be an odd number of incoming, unique, flows to a particular IP address.

As shown in figure 6 there is no evidence that any DDoS attacks occured withing this timeframe, ruling out a DDoS attack as a source of the flows to the IoT device.
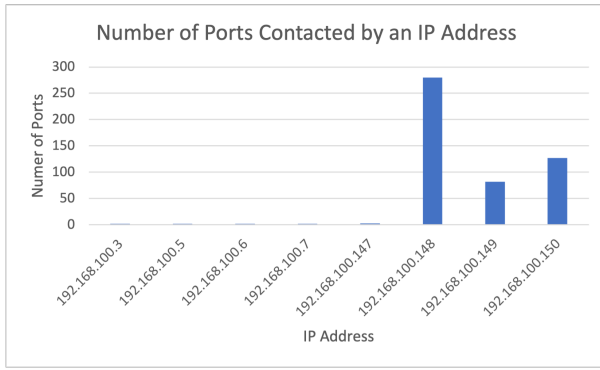
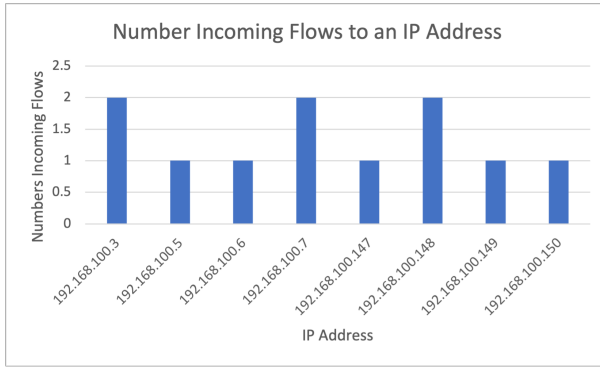Fig. 5. The number of ports contacted per IP address

Fig. 6. The number of flows from unique IP addresses coming into a particular IP address

V. DISCUSSION

Clearly the work done here is just the beginning, however the results already gained show great promise of revealing important information. From so little data and simple queries we can already determine whether two important attacks occur. However the results found by these queries on this dataset are limited. For one, only simple queries were and more advanced queries could and should be used in the future in order to solve more complicated problems and answer more complicated questions. Secondly, the fraction of the dataset used only involved 1,000,000 flows. In order to get a more accurate picture of a network, millions more flows so be gathered and analyzed. Luckily, due to the speed of graph databases, this shouldn't be a problem for similar experiments in the future.

VI. FUTURE WORK

The work in this paper should be considered a beginning rather than an end. There are so many doors and paths the author was not able to take during this project due to various limitations and constraints. Work for both the author and readers for the future include, but are not limited to, expanding the queries performed to answer more complicated questions. In addition, additional work should be done to analyze these basic questions at scale, whether that be scale of time or scale of data. More work also needs to be performed on

the comparison of the utility of graph databases for these tasks in comparison to more traditional methods as well as the benefits and/or downfalls to increasing efficiency of the existing systems for these tasks.

VII. CONCLUSION

Graph databases proved to be a useful tool for IoT device timeseries data. It is both fast to query and the information gained provide meaningful answers to interesting questions.

REFERENCES

[1] Danny Yuxing Huang et al. "IoT Inspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices at Scale". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4.2 (June 2020). DOI: 10.1145/3397333. URL: https://doi.org/10.1145/3397333.

[2] Rohit kumar Kaliyar. "Graph databases: A survey". In: *International Conference on Computing, Communication & Automation.* 2015, pp. 785–790. DOI: 10.1109/CCAA. 2015.7148480.

[3] Nour Moustafa. *The Bot-IoT dataset.* 2019. DOI: 10. 21227/r7v2-x988. URL: https://dx.doi.org/10.21227/ r7v2-x988.

[4] Danh Le-Phuoc et al. "The Graph of Things: A step towards the Live Knowledge Graph of connected things". In: *Journal of Web Semantics* 37-38 (2016), pp. 25–35. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem. 2016.02.003. URL: https://www.sciencedirect.com/ science/article/pii/S1570826816000196.