JHU IDS Module 6 Lab

Audrey Long

07/04/2020

Introduction

This week's assignment is centered around using Neo4j, a popular graph database, to perform some basic IDS-related triage. Neo4j is an immensely popular, scalable database that uses a query language called Cypher. Cypher is pretty neat actually, it's based on English words and ASCII art. We hope you enjoy learning a new tool!

Installing Neo4j

For this assignment there are a number of ways you can run Neo4j. The first option is to run Neo4j in your browser: https://neo4j.com/sandbox-v2/ or http://console.neo4j.org/
Alternatively, you can run a Neo4j Docker container found here:
https://hub.docker.com/_/neo4j or by doing a 'docker pull neo4j'. It's a nice, lightweight option
that you should be able to run successfully inside your VM.

Finally, you can follow these instructions and install it on your machine manually
https://neo4j.com/docs/operations-manual/current/installation/linux/
Whichever method you prefer is fine.

For this Lab i decided to use the browser option https://neo4j.com/sandbox-v2/

Building the Model Network

A Cypher file has been created for building a network containing 9 hosts, two firewalls, and two routers. Which you can find here: https://github.com/jkovba/jhu-sp19-ids/blob/master/neo4jnetwork.cypher .

Run this code inside of your Neo4j installation and it will build the corresponding graph for you.

Deliverables

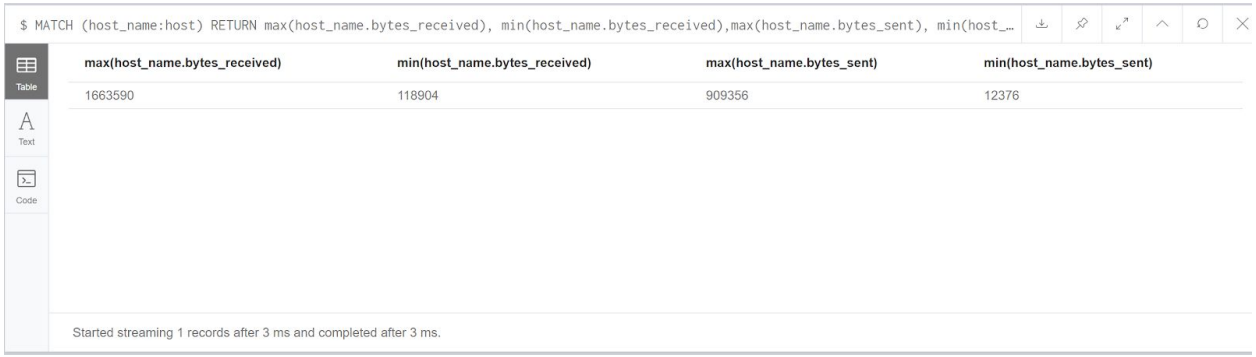
1. Start by writing a Cypher query for calculating the number of nodes in the graph. As a hint, you can use the CONNECTS TO relationship as part of your query. Please provide the query along with your answer.

Where the query MATCH (n) RETURN count(*) returned the number of nodes which is 13

2.a) Next, write a Cypher 1 query for calculating the minimum and maximum bytes sent and bytes received values for the hosts. Provide the query along with your answer and explain how you might use these numbers to see which host is malicious and which one is infected. Justify your choices and explain what other information you would incorporate into your graph to strengthen your analytic.

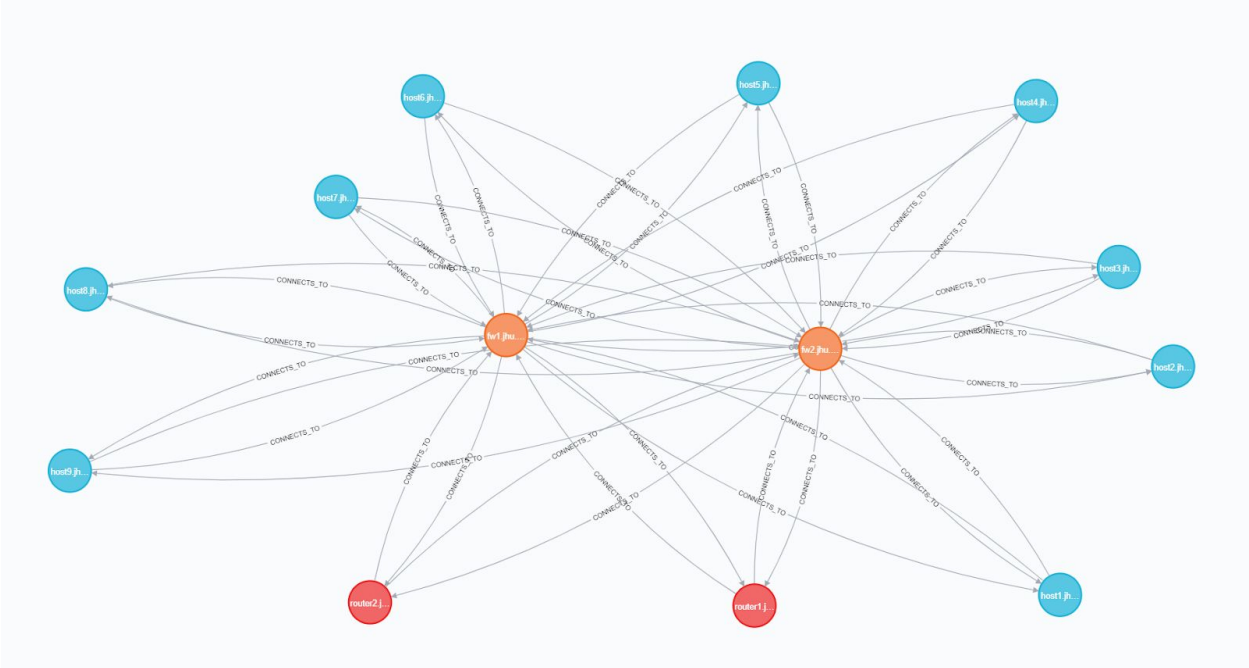
The following query was used to get the information provided below:

MATCH (host_name:host)
RETURN max(host_name.bytes_received),
min(host_name.bytes_received),max(host_name.bytes_sent), min(host_name.bytes_sent)



In general you can use the information of bytes_sent and add to the query to return the IP address which sends out so many bytes. With that information you can easily figure out which host is infected. In general a suspicious looking network will send out many bytes of data to fully infect the rest of the network, or send information back to the malicious user. With the bytes received you can tell which hosts are infected by the sheer number of requests being ingested by that host. TImestamp data could be a very strong indicator of malware infection, for example if tons of packets are being sent out in off hours. The strategy would take advantage of the fact that malware invaders need to communicate with their command and control computers, creating network traffic that can be detected and analyzed. Having an earlier warning of developing malware infections could enable quicker responses and potentially reduce the impact of attacks, the study's researchers say.

3. Finally, please provide a sketch of the actual network topology being represented in the graph.



References

https://www.sciencedaily.com/releases/2017/05/170522081530.htm