IoT Sentry

Class:
CSE 485

Team Name:
Enterprise IoT Passive Mapper

Members:
Nicholas Pinedo - nmpinedo@asu.edu

Kushagra Kshatri - kkshatri@asu.edu

Megan Beaudoin - mbeaudo1@asu.edu

Benjamin McLemore - brmclemo@asu.edu

Jacob Kenny - jakenny@asu.edu

Allen Olesen - amolesen@asu.edu

Sponsor:
ProDefense

Project Description:
An all-in-one passive IoT and embedded device network mapper

Table of Contents

Project Description 3

User Overview 5

Requirements 6

Preliminary Results 7

Conclusion 9

Project Description

The purpose of the project was to create a passive network mapper for IoT devices. To explain what this is, we must first define traditional network mapping. Traditional network mapping entails continuously sending packets from one node to all other nodes on the network. While not guaranteed to map the entire network, this process will map at least the nodes directly connected to the one sending packets. There are several issues with this method, though:

1. The process is time-consuming
2. The process is resource-intensive, meaning the network will either slow down drastically or crash altogether
3. The process is labor-intensive; the person mapping the network must first be trained to do so

Thus, our project sponsor seeks to create a passive network mapper. Instead of sending packets to all other nodes, a passive network mapper observes packets as they traverse the network. It is similar to a man-in-the-middle (MITM) attack in that the packets are not altered or tampered with. Instead, the “mapping node” reads the packet and, using the metadata, determines which nodes are communicating with each other. It then uses this information to construct the network. A passive network mapper, while also not guaranteed to map the entire network, solves each of the aforementioned problems associated with traditional network mapping. The process is passive, meaning it runs in the background and largely unnoticed. Since it is only observing and tracking, it is not resource-intensive and does not affect the network at large. Finally, there is no special training required, as building the network is now automated.

With that in mind, these are the deliverables the project sponsor and our team agreed on:

1. Software to analyze packets (in the form of Packet Capture, or pcap, files) and obtain metadata, such as device brand, device make, device model, etc.
2. Software to do (1) in real time instead of relying on previously-captured pcap files
3. A visualization tool to represent the network we had mapped thus far
4. An app enabling the end user to access (2) and (3) simultaneously

How we set out to achieve these and our progress in doing so is detailed later in this document.

User Overview

The stakeholder for our enterprise IoT passive mapper is ProDefense. ProDefense is a cybersecurity company, based in the Phoenix metropolitan area, that provides security services to a range of companies in almost any industry. Their team offers many services such as pentesting for IoT and embedded devices, web application pentesting, firmware security, and many more. Their primary mission statement is to provide the necessary security for a customer no matter their needs. They emphasize that if a service they have does not perfectly align with a customer's desire, the customer should contact ProDefense directly. They also hold many cybersecurity certifications from top security organizations to further prove their credibility.

ProDefense also is a contributor for research projects including open-source and community projects. This research is displayed in various blog posts which cover a multitude of topics in efforts to educate and help the community with security. ProDefense has also found new vulnerabilities and has documented them in the CVE glossary. CVE stands for Common Vulnerabilities and Exposures. It is seen as the gold standard and central hub for tracking, logging, and documenting known cybersecurity vulnerabilities and attacks. ProDefense also has made various security tools such as Spoofy, a program that checks if a list of domains can be spoofed. Most of their research, work, and passion ss focused on improving security and finding vulnerabilities for IoT and embedded devices.

Scenarios

The Enterprise IoT Passive Mapper can be used on a network as a defensive security tool. It would serve to make network topology clear and notify if any unauthorized or unexpected devices are on a network. A user mapping their own network would be most interested in how the network mapper displays the connections between their devices, and which connections are most readily visible to the outside observer. They could see what devices are flagged as most

likely to be IoT devices and take measures to conceal that on their own if they deem it the best course of action.

The IoT Passive Mapper can serve as a network recon tool when pentesting. Just as it would help for network topology defensively, it can also serve as a tool to discover and map vulnerable IoT devices. A user that planned to use the IoT Passive Mapper technology in this manner would specifically want to know which devices are connected to the network and which ones are likely to be IoT devices. They would want to know as many details about the connected devices as possible, in order to take advantage of their unique vulnerabilities.

Requirements

- Must be able to perform without actively interacting with the network
    - One of the values of this application is its ability to monitor and analyze network traffic without alerting the network of its existence; otherwise, it fails as a security tool because if bad actors know that it is present on the network, they can take measures to avoid getting detected by it.
- Must be deployable on many environments
    - Because this software must be able to monitor different types of networks, and because it will be used by different entities with different resources to run it, it must be able to be deployed in many different environments. This increases its value, since it means that most entities will be able to use it, not just those which already have a supported setup.
- Must have a distributed architecture
    - The application must have a distributed architecture because it will need the capacity to run in different types of networking environments and because the user may desire to replicate some components on multiple systems. This allows the application to provide value both to users intending to run the application on a single machine and to users running different pieces on different parts of the network. It also gives users the ability to monitor multiple networks by duplicating only some parts of it, rather than the whole system.
- Cannot use proprietary software
    - IoT Sentry is meant to be free, open source software. This provides value by allowing companies to use the software for free and giving them an option that avoids vendor lock-in.
- Must have a user interface to interact with the application
    - In addition to network administrators, this software will also potentially be used by other individuals within an organization who may not be as technologically literate. As such, they need a way to interact with the application that is generally accessible and provides an intuitive user interface. This provides value by allowing organizations that use the software to hire separate individuals with different skill sets to maintain and use the software, so that each job is being performed optimally by someone with the relevant skills.

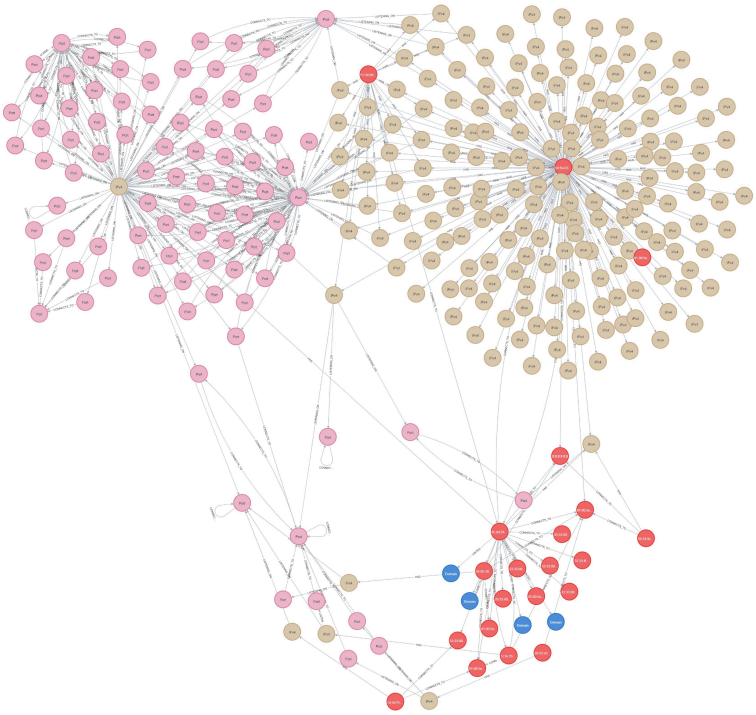
- Must use a Github repository for the project
    - Because IoT Sentry is going to be free and open source, it needs to be maintained in a publicly accessible repository where anyone can access the application and pull changes into it. This provides value because it allows the application to be easily passed on to successive groups of people working on it and to be maintained by the software community at large once it is released.

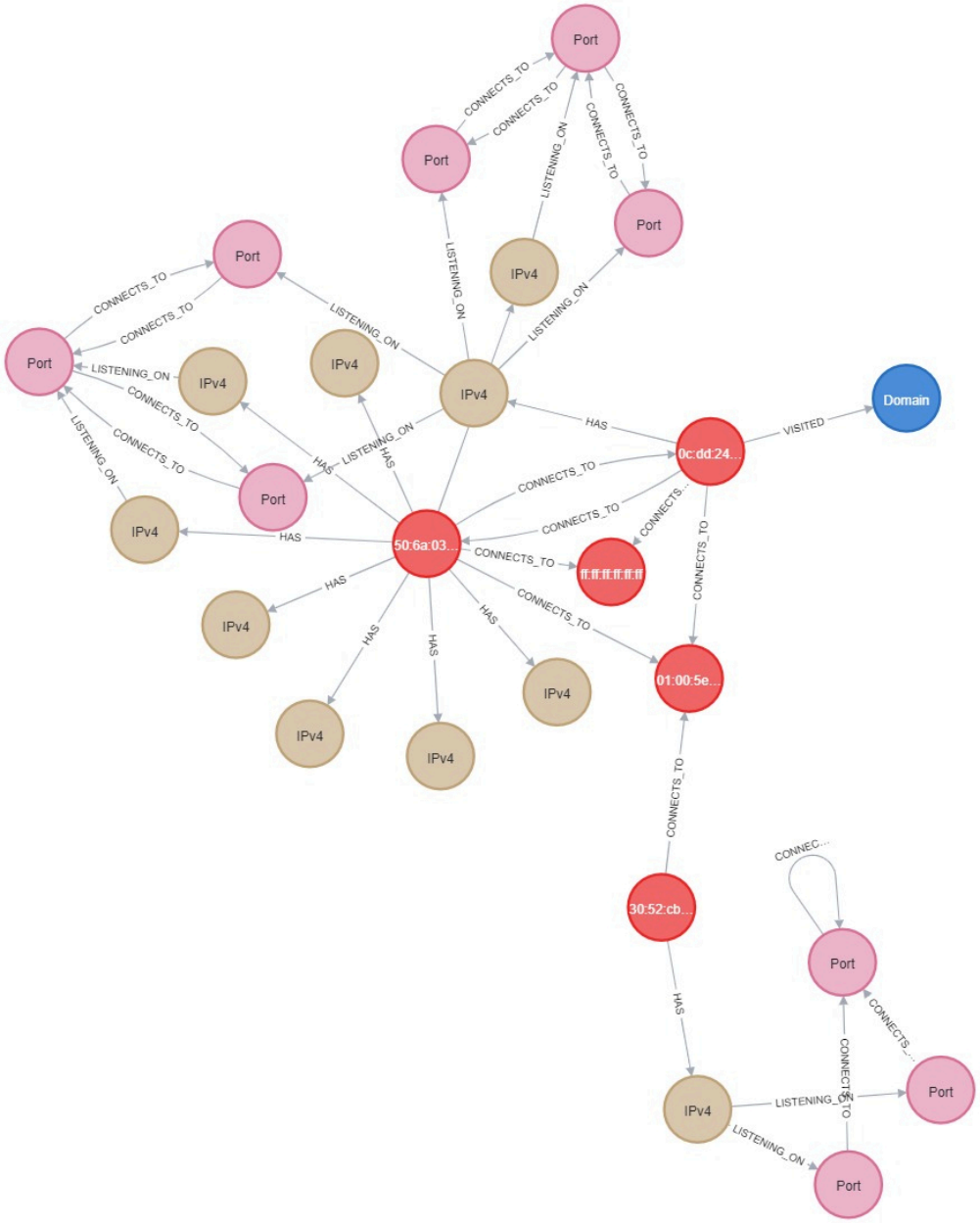
Preliminary Results

A test of our MAC address prefix match can be seen in action here: https://www.youtube.com/watch?v=Q3xVLjYoTdI

In this example, we show how our matcher works on two separate pcap files captured by team members for testing. It takes Zeek logs and processes them to be an ingestible format for both the database and other python scripts. It then takes these logs and performs the MAC address matching on them before outputting that information also in a json format that can be ingested easily by any other script that needs this data or the database.This has been tested against both valid and invalid PCAP files. Our sponsor has given us good feedback regarding this script as our minimum results they expect are identification by vendor. This gives us a fantastic starting point for identifying this. However, this is not an absolute identifier and only exists as a stepping stone.

Another area that we have had good results is our data visualization. On the dashboard we plan to show a visualization of the network data and communication over the wire. Here, we use neo4j to visualize one of our test case PCAP files. Here is the full image and a zoomed in version:





These graphs allow the user to see the connections and nodes in a digestible way so they can check the network for any suspicious devices. It will aid our future endeavors by making it easier for the user to help the program identify what types of devices the user wants on the network and how they act.

Conclusion

Our IoT Passive Mapper is on the right track to completion. We have considered the potential users for the project and have worked with our sponsors to determine the specific goals and project requirements for our product. We have made substantial progress on our graphing technology, and with more research on other potential identifiers for IoT devices, we will be able to add more functionality so our IoT Passive Mapper can be even more accurate.

In conclusion, our project to build a Passive IoT Device Mapper proved to be a challenging and rewarding experience. We were able to research different topics associated with this project and gain an insight into the complexity of building a comprehensive IoT device mapper. We were able to research containerization parameters, Zeek, databases, Flask, and ELK stack. We were also able to build a graph database and a MAC address prefix match program. We plan to continue working on this project by integrating all the components together to build the final product. We are confident that our project will benefit network security researchers and administrators by providing them with an efficient way to detect and analyze IoT devices on their networks.