Riley Schina
William Zhongjun

Milestone 2

App Summary: *Lock & Key: Network Check*

The purpose of this app is to be a simplified way for people to better secure themselves in an increasingly online world. By focusing on making it easier for people to check their own networks and devices and find relevant information to help meet their network security needs.

Lock & Key: Network Check, is an app that searches for local machines on your network and helps to identify vulnerabilities within the system, then directs the user to resources to help.

Progress Report: Completed

User authentication using Firebase, with the email and password mode being what we selected for implementation.

Complex Data storage of network searches using Firebase, storing the information as database items. Data is accessible per user, and contains information on the public device IPs, time of search, and geoLocation of the devices; as well as CVE search terms. Previous searches are available to view in the Logs screen of the app.

Local AsyncStorage was implemented to remember app settings, specifically as the controller for the light and dark modes of the app. This was used since the information is lightweight and made no sense to store off device. The ability to toggle between Light and Dark modes has also been implemented.

Drawer Navigation was selected as the main navigation of the app, as we felt that it made for a better experience in regards to accessing the different screens.

The project mainly uses the nmap, Axios (using CVE information from NIST),Google geolocation, and Google maps APIs to make the security scans, search for security terms, and compare the locations of devices to the user, respectively. The nmap checks for the public IP information of devices, and feeds the data into the CVE database by using Axios as a bridge. The geolocation API is used to compare the local GPS location of the device to the devices found using the nmap scan, and displays the locations using the Google maps API.

Progress Report: In Progress

While most of the UI has been completed, there are some areas for improvement, like in our settings screen, which currently only has the dark and light mode toggle. Further UI improvements need to be made to display the large amounts of CVE information, as it currently displays as large paragraphs in a flatlist. We have so far used colour coding of the various parts of the CVE to improve readability, however we need to better separate each flatlist item.

Challenges and Solutions

Challenges: nmap -sV only detects the version of the service running on port; doesn't output enough information for NIST CVE lookup to output useful information.
Solution: used nmap -A to detect MAC, OS, version, etc, implemented -T4 to speed up scans; kept fast vs advanced scan option for -A and -sV, updated and implemented 2 versions of parsing logs

Challenges: nmap doesn't work with reverse IP geolocation as current implementation relies on the subnet mask and IP
Solution: used MAC/IP reverse loop up API to reverse IP geolocation and obtain information, ISP information, proxy/mobile detection

Challenges: nmap doesn't return internet IP, only subnet IP, which can't be used in IP geolocation.
Solution: Used api-ipify.com to obtain IPv4, as IPv6 affects the IP geolocation API's accuracy

Challenges: google maps needs latitude and longitude to display log scan locations
Solutions: store lat and long in firestore log, updated and mutated praising function to display useful data in google maps integration

Use Case

A penetration tester wants to enumerate and scan the target within the rules of engagement. With management and leadership observing, he wants to present the scout result in a human-readable form. Due to the fact that CVE and nmap or other penetration/network testing tools are available only in a command line interface, he uses our app to automate scanning and help explain with visualisations to his employer.

Project Requirements

Requirement	**How it was met**
User Authentication	Firebase, email and password login
Simple / Local Storage	Async storage of settings, light and dark mode
Complex / Database	Firestore database, storing public IP, location, time data, etc. Retrieval of that data in the logs screen
API Integration	Nmap, Axios, Google geolocation, Google maps
UI and Navigation	Styling mostly finished, Drawer navigation implemented
Native Feature	GPS used to compare the location of network devices to the user's device.