Brief Code Report

How the code works

The log file analysis code works by reading into the .csv CPU usage file created from another code set and determining which values are above 70%. If a CPU value is above 70%, it sends a debug warning message saying that the CPU usage is too high and needs attention. The next set of code we have is the Logging test code, which logs the computer's current CPU and stores it in a .csv file. It monitors the CPU usage every second and records the current timestamp, CPU usage, and a fixed hostname. Each data is added into a new row in a dataframe and appends the data to a CSV file called cpu_usage.csv. It then prints the timestamp and CPU usage to the console every second. Next is the Nmap code. The code starts with a function taking in a single argument, which is the target/the IP address going to be scanned. The IP address being scanned is the localhost / 127.0.0.1. The results of the scan which are open ports and services are captured and printed. This is useful for network diagnostics, security auditing, or discovering services running on a machine. The last set of the code is the scapy packet sniffing code. Scapy is used to capture network packets. The function monitor_packets checks if a packet contains both an IP and a TCP layer. For each matching packet, the function extracts and prints the source and destination IP addresses. The sniff() function captures 10 packets in total processing each using the monitor_packets function. This script is useful for network monitoring or basic traffic analysis, helping to track the source and destination of TCP packets on the network.

Implementation

Each set of code was scripted using the Python language. Each set of code is relatively universal, meaning it can be run across any computer as long as the necessary tools are downloaded. For the most part, most of the codes needed tools such as Panda, time, subprocess, nmap, and scapy. While panda is downloaded and installed through the python language, tools such as nmap and scapy have to be manually downloaded and imported through the system's path, meaning we had to interact with the system command line in order to get it working.

Findings

For the findings in the log analysis code, we found that when CPU usage was above 70%, a warning debug message would be sent out. This is useful when an anomaly occurs with the CPU usage, it brings our attention to it. With the findings for the log testing, we simply found the CPU usage for the current system that the code was being run on, as well as the time stamp and the fixed hostname. This can be useful for monitoring a simple system performance attribute, logging it, and storing it for future use in case of an anomaly occurring later on. With the Nmap system scan code, the scan revealed that 3 ports(22, 80, 443) are open on 127.0.0.1 and are running services like SSH and Apache HTTPD. The scan shows which versions of these services are running, which could be useful for identifying vulnerabilities. The host 127.0.0.1 is up and reachable, and Nmap successfully completed the scan in less than a second, indicating that there is minimal network latency. Last but not least, the scapy code prints the source and destination IP address of 10 packets that contain both TCP and IP layers. The output gives insight into the devices communicating over the network and the flow of traffic. This information is useful for network monitoring, troubleshooting, or understanding traffic patterns in a local network or on the internet.