Brief Report on the assignment:

The Purpose of each script and how it was implemented:

1. Log file analysis:
Log file analysis scans a system log file for entries that are suspicious and are often classified as unauthorized or failed. What this does is make a report of these suspicious activities.

It is implemented by opening a specific log file. This log file must match the same name as your code. We named our log file “system_log.log” as the example, so our code had to have the same name. Also, note that the script and the log must be within the same folder. Moving on, we mentioned before that we filtered it by specific keywords; this would open a new file called “summary_report.txt”; it lists every suspicious activity when you run the code.

2. Systems Performance Monitoring:
Systems Performance Monitoring monitors the CPU and memory usage and logs the data. If the CPU goes above a defined threshold, an alert is triggered.

It is implemented by importing the “psutil” library. This checks the system’s current CPU and memory usage when you run the code. If the CPU exceeds the 90% mark (which was set by me), an alert is printed, and a log of both the CPU and memory is saved in “performance_log.txt.”

3. Alert Generation via Email:
An alert generation purpose is to notify the system admins via emails in the event of a critical performance issue, specifically a high CPU usage.

It is implemented by importing “smtplib” library and using Gmail’s SMTP server. The “send_alert” command creates and sends an email if the CPU usage is above the threshold, which is 90%. The email contains the alert message from a designated sender to a recipient.

4. Security Automation – Nmap Network Scan:
This network scan aims to identify open ports and services running on a given network target, which facilitates and helps find vulnerabilities.

It is implemented by using the “subprocess” module, and the “run_nmap” function runs a Nmap scan. The “-sV” option identifies open ports and vulnerabilities on the target IP which is “127.0.0.1” in this case.

5. Security Automation – Packet sniffing with Scapy:

The purpose of Scapy is to capture and analyze network packets, mainly focusing on IP and TCP layers as it monitors suspicious network activity.

It is implemented by “scapy”, and the “monitor_packets” functions listen on a network and captures ten packets. The source destination and IP addresses are displayed for each packet containing IP and TCP layers.

Results:

1. Log file Analysis:

Suspicious entries, such as failed attempts or unauthorized success warnings, were successfully logged in “summary_report.txt.” It gives an exact count of the entries that could help an potential investigation.

2. System Monitoring and Email Alerts:
When testing the script, CPU and memory usage were logged, providing insights into the system performance trends. If the CPU usage exceeds the threshold, an alert and email notification will be triggered. This shows that the alerting system is effective.

3. Nmap Network Scan:
The localhost scan “127.0.0.1” lists open ports and services. This would help with identifying potential vulnerabilities.

4. Scapy Packet Sniffing:
The captured packet information displays the relevant traffic’s source and destination IP addresses, analyzing real-time traffic analysis. It is helpful for identifying intrusion attempts.