Weekly Summary Report: Network Traffic Monitoring and Intrusion Detection

Date: 19/10/2024

Objective: This week, the focus was on setting up and using Snort for network traffic monitoring and intrusion detection, using tools like Wireshark for packet analysis, and writing custom rules to detect common network attacks.

Tasks Completed

1. Network Security Monitoring Basics:
   Learned about monitoring network traffic for detecting anomalies, including familiarizing with tools such as Wireshark and Snort.
   Investigated key concepts such as packets, protocols (TCP, UDP, ICMP, etc.), and traffic analysis for identifying unusual network behaviors.

2. Wireshark Setup and Packet Analysis:
   Installed and configured Wireshark to capture network traffic, focusing on protocols like HTTP, DNS, and ICMP.
   Practiced using filters in Wireshark to isolate and analyze specific types of traffic and detect anomalies like unusual IPs, ports, and protocols.

3. Snort Installation and Configuration:
   Installed Snort on a virtual machine and configured it for intrusion detection.
   Wrote basic Snort rules to detect attacks such as port scans, ICMP floods, and HTTP traffic on nonstandard ports.
   Tested Snort's functionality by generating traffic (e.g., port scan with `nmap` and ICMP flood with `ping`) to simulate realworld attacks.

4. Traffic Monitoring with Snort:
   Configured Snort to monitor network traffic and log alerts in humanreadable format.
   Simulated common network attacks (port scan, ICMP flood) and analyzed Snort's alerts, documenting the detected threats and their potential impact.
   Generated reports based on the analysis, including details about the nature of attacks and suggested mitigations.

Challenges Faced

1. Snort Rule Writing:
   Writing Snort rules required careful attention to detail to avoid false positives. Some of the initial rules generated unexpected alerts, requiring finetuning of thresholds and conditions (e.g., adjusting the number of packets in port scan detection).

2. Traffic Overload:

Analyzing network traffic captured with Wireshark was challenging when there was a high volume of packets. Identifying relevant traffic amidst the noise required refining filters to focus only on the most relevant protocols or ports.

3. Performance of Virtual Machines:
Running Snort and Wireshark on virtual machines resulted in performance issues when generating large amounts of network traffic. This slowed down analysis and required managing system resources effectively to maintain monitoring.

Improvements for Future Monitoring Setups

1. Rule Optimization in Snort:
Improve Snort rule efficiency to avoid unnecessary alerts and ensure more accurate threat detection. Focus on refining thresholds and customizing rules to suit the specific network environment.

2. Better Traffic Capture with Wireshark:
Use Wireshark's advanced filters and protocol analysis features more effectively to capture only relevant traffic. Prioritize capturing critical traffic over largescale general packet captures.

3. Scalability Considerations:
For larger networks, consider scaling the monitoring setup by using multiple sensors with Snort to distribute the load and ensure better coverage of the network.

4. Automating Report Generation:
Develop scripts or use tools like `Barnyard2` to automate the parsing and analysis of Snort logs, generating regular, standardized reports on detected threats. This would reduce manual effort and provide timely insights into network health.

5. More InDepth Attack Simulation:
Test the monitoring setup with more complex attack scenarios (e.g., DDoS, malware traffic, SQL injection attempts) to ensure the system is prepared for a wider range of threats.

Conclusion

This week was focused on learning and implementing essential network monitoring and intrusion detection practices. The key tasks were successfully completed, including setting up and configuring Snort and Wireshark, writing basic intrusion detection rules, and analyzing network traffic for potential threats. Although there were challenges with rule writing and traffic overload, these were useful learning experiences that will help improve future monitoring setups. Going forward, refining Snort rules, improving packet capture techniques, and automating report generation will be key areas for improvement to enhance the effectiveness of the network monitoring process.