Cybersecurity Lab Journal – #3

Lab Description:

In this lab, I worked with Suricata, an open-source intrusion detection and prevention system. The goal was to learn how rules are written, how they generate alerts, and how to read Suricata's log files. I used a provided sample .pcap file and a custom.rules file to trigger alerts and then examined the results in fast.log and eve.json. This helped me understand how IDS rules detect specific network activity and how those alerts appear in different log formats.

Date: Record the date of the journal entry.	**Entry:** #3 **Date:** October 12, 2025
Description	I reviewed and ran a custom Suricata rule against captured network traffic to see how alerts are generated. I also practiced using terminal commands to view and format Suricata's output logs.
Tool(s) used	Suricata, Bash terminal, `jq` (for formatting JSON logs)
Key Findings / Skills Practiced:	• Learned the structure of a Suricata rule (action, header, options). • Used the rule: `alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:established,to_server; content:"GET"; http_method; sid:12345; rev:3;)` • Triggered alerts using `sudo suricata -r sample.pcap -S custom.rules -k none`. • Located logs in `/var/log/suricata/` and examined both `fast.log` and `eve.json`.

	• Used `jq` to format JSON output and extract fields like timestamp, flow_id, alert signature, and destination IP. • Verified that the alert signature **"GET on wire"** appeared in both log files. • Identified the destination IP **142.250.1.139** and confirmed severity level **3** for the first alert.
Important Log Outputs:	**Important Log Outputs:** • `fast.log` showed short alert summaries including the rule message and IP flow. • `eve.json` included full JSON-formatted event data, including alert details, flow IDs, and severity. • Example event: `["2022-11-23T12:38:34.624866+0000",14500150016149,"GET on wire","TCP","142.250.1.139"]`
Additional notes:	This lab helped me understand how Suricata rules actually work when monitoring network traffic. Seeing how one rule triggers alerts across multiple logs made it easier to connect the rule logic to the network data. It also showed how JSON logs are better for deeper analysis than the older `fast.log` format.
