White Paper

How Corelight Accelerates Incident Response with Zeek and Suricata

Introduction

Critical security questions often go unanswered when alerts fail to provide the data needed to validate and investigate them. That's why many of the world's top blue teams deploy both the open-source Zeek network security monitor and open-source Suricata IDS in parallel. Suricata excels at raising alerts via binary pattern matching and Zeek excels at generating rich, connection-linked protocol logs that empower analysts to make fast sense of network activity around a given alert.

With both signal and evidence security analysts can move quickly and decisively to respond to high severity threats. Zeek's evidence gives incident responders clear answers to the critical questions that arise during investigations and help analysts distinguish between signal and noise. Threat hunters rely on Zeek's rich network evidence to rapidly test their hunting hypotheses and often turn to Suricata to transform their discoveries into automated threat detections via new Suricata rules.

Implementing these two open-source security technologies on a shared sensor, however, can be challenging, especially in high throughput environments. Sensor deployment and tuning can take months, performance and packet loss problems can arise at scale and analysts must ultimately manage and correlate two distinct datasets. Corelight's physical sensors now enable organizations to easily deploy this open-source design pattern in a single form factor that accelerates security analysis via Zeek/Suricata data unification, scales traffic analysis via a unique shared-CPU architecture, and takes just minutes to deploy.

With Corelight, analysts get alerts and the data they need to investigate them.

A Unified Dataset to Accelerate Investigations

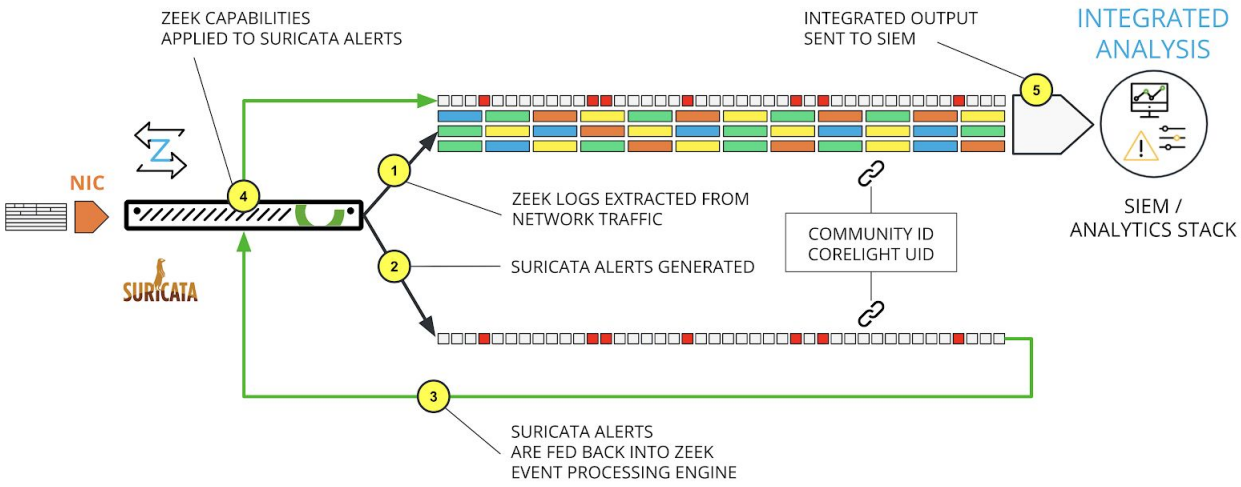

Diagram 1: Corelight Sensor - Zeek and Suricata data unification

As depicted in Diagram 1 above, the Corelight Sensor's high performance NIC ingests mirrored traffic via a packet broker, span port, or optical tap, whereupon:

1. Zeek transforms the packets into rich, connection-linked Zeek protocol logs
2. Suricata inspects packets in parallel for signature matches
3. Suricata alerts then pass *back* through Zeek's event processing engine
4. Suricata alerts become their own distinct Zeek log with a Connection UID
5. The unified dataset is streamed to a SIEM

Corelight generates a proprietary "Suricata_corelight" log for each Suricata alert, but can also support the "Suricata_eve" format if customers running Suricata prefer to maintain an existing security data pipeline that requires the native Suricata alert format. The sensor supports Suricata version 5.0.2 and can process tens of thousands of rules, though Corelight does not supply rulesets in its product so customers must develop their own custom rules and/or load 3rd party rulesets on the sensor such as those offered by ET Pro, Crowdstrike, and Talos.

Corelight's "Suricata_corelight" alert log notably includes Zeek's unique connection ID (UID), which provides a superior flow identifier compared to the open-source Community ID standard, as the UID is unique and the Community ID's hashing function of the 5-tuple is susceptible to potential collisions.

Where open-source implementations of these tools generate two distinct datasets, Coreligh's approach of embedding Suricata alerts directly into the Zeek logging framework simplifies downstream data export and management and removes the need to manually pivot between the two datasets in a SIEM. Consider the following example, where in the course of investigating a Suricata_corelight alert an analyst must find portable executable files coming from a TCP session where the HTTP "host" header was an IP address and then make a true positive or false positive assessment of the initial alert:

Suricata Rule:
alert http $EXTERNAL_NET any $HOME_NET any INFO SUSPICIOUS Dotted Quad Host MZ Response"; flow:established,to_client; flowbits:isset,http.dottedquadhost; file_data; content:"MZ"; within:2; content:"PE | 00 00 | "; distance:0; metadata: former_category INFO; classtype:bad-unknown; sid:2021 076; rev:2,; 201 updated_at 2015 05_07;)

Suricata_corelight alert:
{"_path":"suricata_corelight","_system_name":"sensor.lan","_write_ts":
"2020-04-17T17:13:49.872575Z","ts":"2020-04-17T17:13:49.868945Z","uid":"C12kb511xh3c
ZEwS2g","id.orig_h":"10.3.11.194","id.orig_p":49816,"id.resp_h":"64.44.133.131","id.resp_p"
:80,"suri_id":"SYAmHDogttA5","service":"http","flow_id":1563719810015877,"tx_id":1,"pcap
_cnt":01"alert.action":"allowed","alert.gid":11"alert.signature_id":2021076,"alert.rev":2,"ale
rt.signature":"ET INFO SUSPICIOUS Dotted Quad Host MZ Response", "alert.
ure_id":2021076,"alert.rev":2,"alert.signature":"ET INFO SUSPICIOUS Dotted Quad Host MZ
Response", "alert. category":"Potentially Bad Traffic" ,"alert.severity":2,"
community_id":"1:8RrKhTjZLxLyr/AuWD1216wgoHw="}

Via Corelight's Connection UID highlighted in green an analyst can pivot directly from the Suricata_corelight alert into Corelight's files.log, see the file's MD5 hash and validate it as malicious on VirusTotal. In the open-source implementations, this would require an extra pivot from the Suricata alert to the Zeek conn.log via the Community ID, which runs a non-zero risk of flow ID collisions.

When an individual analyst processes hundreds of alerts per day this small, but significant efficiency gain per investigation can result in meaningful aggregate time saved that can be reinvested in high priority investigations. And for organizations that rely primarily on PCAP to investigate Suricata alerts, Corelight's Zeek-based approach offers even more dramatic efficiency gains in the investigation since Zeek logs are optimized for lightning-fast search and pivots. Compared to the relatively slow processes of deriving insights from manual packet analysis, the difference between investigating with Zeek logs vs. full packet capture/analysis can easily be on the order of minutes vs. hours per alert.

An Innovative Architecture Designed for Scale

Open-source Zeek and Suricta implementations require that engineers estimate the tools' expected workloads and then pin their respective processes to dedicated CPU cores. This static architecture can produce painful bottlenecks when traffic workloads exceed the pre-assigned capacities and can also make packet loss difficult to measure and manage across both tools.

Corelight has pushed the envelope on this design pattern with an innovative architecture that assigns Zeek and Suricata workloads across shared CPUs to deliver elastic performance as depicted in Diagram 2 below:

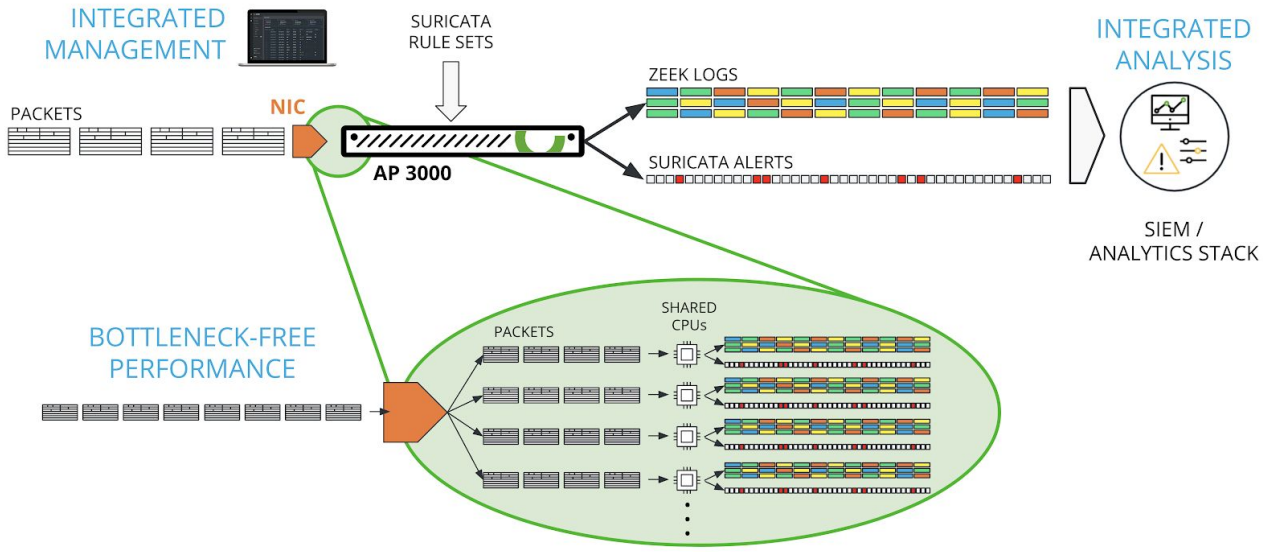

Diagram 2: Corelight Sensor - Zeek and Suricata shared CPU architecture

With Corelight it's not just Zeek and Suricata deployed on a single piece of hardware. We have engineered the two open source projects to truly work together and interoperate, processing packets in parallel on shared CPUs to deliver bottleneck-free performance even in high throughput environments.

Simplified Data Management

Lastly, one of the key benefits of Corelight's sensor design is the integrated management of both Zeek and Suricata data, handled through Corelight's Fleet Manager. Security teams can flexibly filter and export Zeek logs and Suricata alerts via Corelight's management console, streaming Suricata alerts to their SIEM with just a filtered subset of Zeek logs, for example, while sending a complete copy of Zeek logs to cold storage for later forensic analysis.