Snort3 Data Analysis with Elastic stack

Speaker: Josh Sullivan

Welcome to my Presentation
Welcome to my talk on analyzing datasets generated by the intrusion detection system Snort3 and visualized in elastic stack.

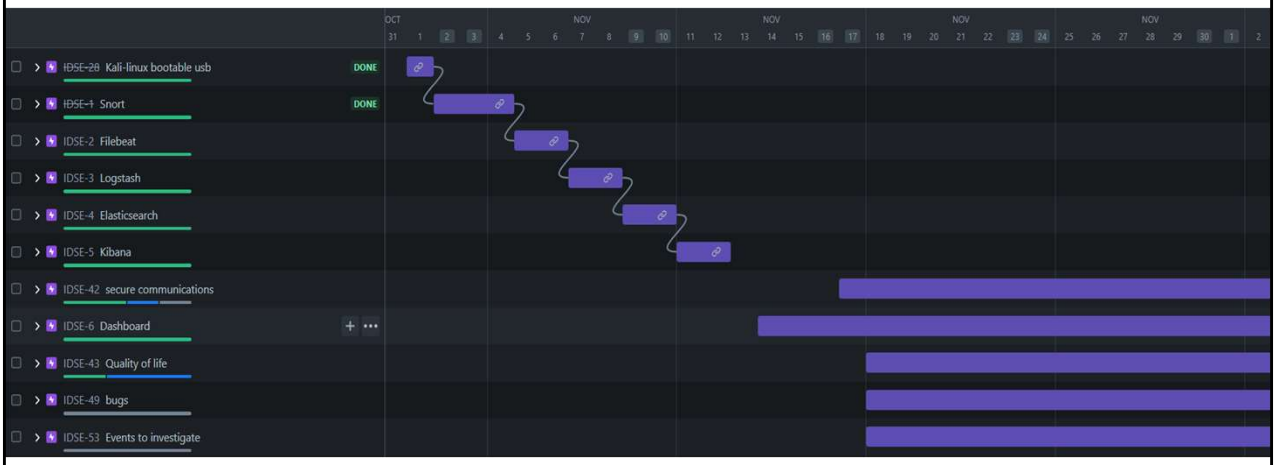
Introduction
Today, I'll walk you through how I used the elastic stack pipeline (Filebeat → Logstash → Elasticsearch → Kibana) to process Snort3 outputs. The primary focus of this presentation is to reflect on the project and present my findings.

Table of contents

Table of Contents

- Jira timeline will provide a little inside in the methodology of the project.
- Architecture should provide an understanding of how the data centralized and visualized.
- Data collection will show us what an alert looks like produced by snort.
- Data dictionary provides how I understood each field.
- Overview Dashboard and Activity Dashboard are the working examples.
- Analysis Process will provide an understanding of the anomaly detection process.

Jira Timeline

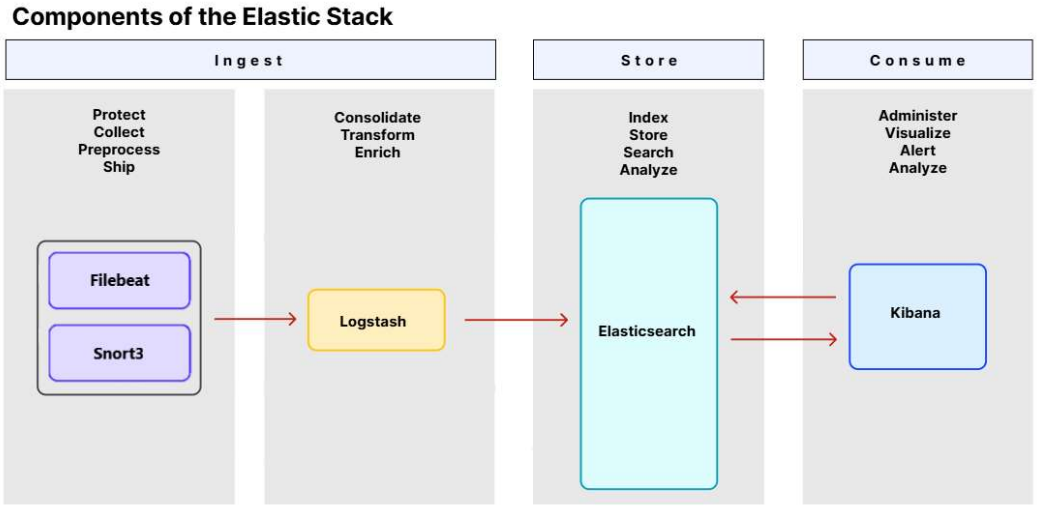


To ensure a structured approach, I started by breaking the project into epics and tasks in Jira. This allowed me to estimate timelines and plan effectively using Jira's timeline feature.
After reviewing the documentation for each component, I set a two-week goal for implementation following the waterfall software development model.

Structure of Epics:
Each epic corresponds to a software component, such as Snort, Filebeat, or Logstash, with recurring tasks focused on:
•Secure communications
•Dashboards
•Quality-of-life improvements
•Bug
•Events to investigate

Due to the dependencies between components, I followed a waterfall methodology. The result was a clear, actionable plan that ensured consistent progress throughout the project.

The Elastic Stack



Ingest -> Storage -> consumption
Elastic stack is group of components that allows ingestion, storage and consumption of log data resulting in the centralization of data allowing for the creation of accessible dashboards in close to real-time.

At the start of the pipeline, we have
- Snort 3 generates an alert in json-format based is predefine internet traffic rules.
- Filebeat configure to a logfile and ships it to a centralized server which in my case is logstash.
- Logstash dynamically processes unstructured data and prepares it using grok patterns before shipping to Elasticsearch for storage.
- Elasticsearch is the core component that centralizes all your data, allowing for various software components to analyze and create visualizations. In my case I chose to work with kibana.
- Kibana is used to create graphs and dashboards to visualize meaningful information

Data Collection

General Information	
Action	Allow
Class	None
Message	(ipv4) IPv4 packet to broadcast destination address
Rule	116:414:1
Priority	3

Source Details	
Source Address	192.168.1.3
Source Port	63341
Source Application	192.168.1.3:63341

Destination Details	
Destination Address	255.255.255.255
Destination Port	6667
Destination Application	255.255.255.255

Ethernet Details	
Ethernet Source	84:E3:42:2F:0C:63
Ethernet Destination	ff:ff:ff:ff:ff:ff
Ethernet Length	214
Ethernet Type	0x800

Timestamp and Interface	
Timestamp	11/29-07:16:41
Interface	/Downloads/raw_Wifi.pcap

To begin, alerts generated by Snort 3 are processed into a structured format, specifically JSON. Each alert captures a wealth of information about network events, including general information, source and destination details, timestamp, and Ethernet details.

Key fields include:

- **rule**: The rule ID that triggered the alert.
- **src_addr**: The source IP address.
- **dst_port**: The destination port targeted by the connection.
- **msg**: A message summarizing the nature of the alert.

To enhance clarity, I've also created a tabular representation of this data to categorize and explain each field in more detail. This provides an accessible view of what an alert looks like and lays the foundation for analysis in later stages.

Data Dictionary

Descriptive Name	Field Name	Format	Data Type	Suitable Graphs	Usage and Purpose
		General Information			
Action	`action`	Text (e.g., "allow")	String	Pie Chart, Bar Graph	Indicates whether the packet was allowed or blocked by the IDS.
Classification	`class`	Text (e.g., "none")	String	Bar Graph, Grouped Chart	Represents the alert classification or threat type.
Alert Message	`msg`	Text	String	N/A	Human-readable description of the alert.
Rule ID	`rule`	Text (e.g., "116:414:1")	String	Bar Chart	Identifies the Snort rule that triggered the alert.
Revision Number	`rev`	Integer	Number	N/A	Version of the rule that generated the alert.
Alert Priority	`priority`	Integer (1-5)	Number	Bar Chart, Heatmap	Specifies the alert's priority level.
		Source Details			
Source Address	`src_addr`	IPv4 Address	String	Geolocation Map, Bar Chart	Specifies the source IP address for the packet.
Source Port	`src_port`	Integer	Number	Bar Graph, Histogram	Indicates the port number on the source side.
Source Application	`src_ap`	IP:Port (e.g., ":60802")	String	Network Graph	Combines source address and port for contextual analysis.
		Destination Details			
Destination Address	`dst_addr`	IPv4 Address	String	Geolocation Map, Bar Chart	Identifies the destination IP address for the packet.
Destination Port	`dst_port`	Integer	Number	Bar Graph, Histogram	Specifies the port number on the destination side.
Destination Application	`dst_ap`	IP:Port (e.g., ":6667")	String	Network Graph	Combines destination address and port for contextual analysis.

The data dictionary serves as a critical tool for understanding the structure and purpose of each field captured in Snort 3 alerts.
Fields are categorized into:

- **General Information**: Provides high-level details like actions and classifications.
- **Source Details**: Captures information about the source of network traffic, including IP address and port.
- **Destination Details**: Provides similar details about the traffic's destination.

For each field, the data dictionary specifies:

- **Data Type**: Whether it's a string, integer, or another type.
- **Visualization**: What type of graph or chart would best represent this data.
- **Usage**: How the field contributes to identifying patterns, anomalies, or security threats.
- **Purpose**: Its role in overall analysis and network monitoring.

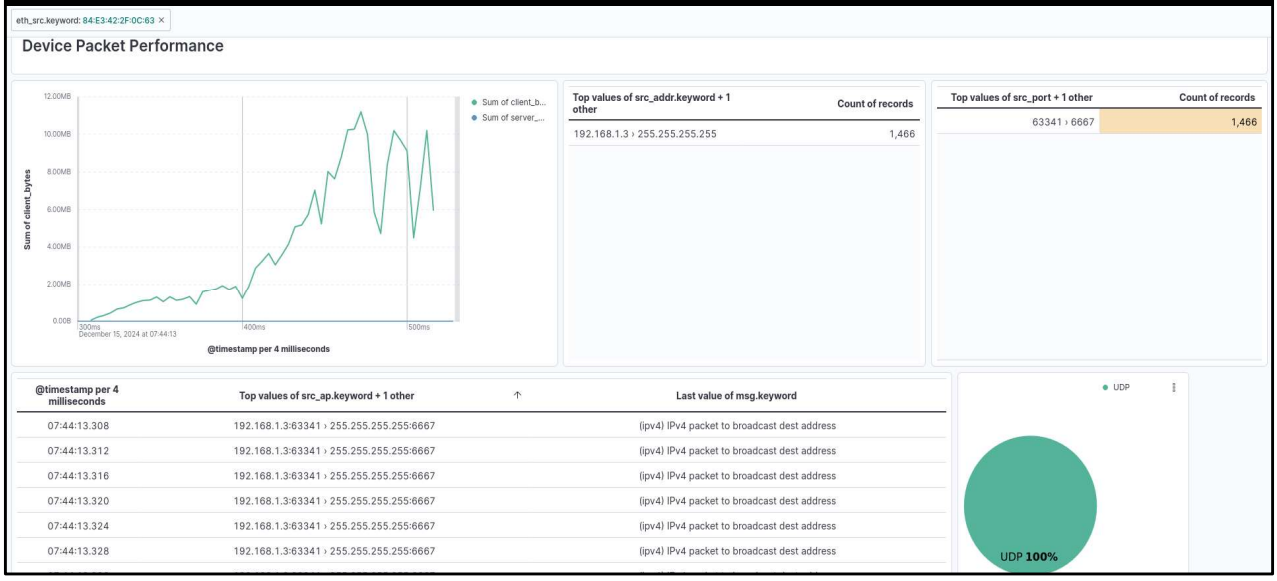
This structured approach ensures consistency and makes it easier to interpret and visualize the data when creating dashboards.

Example: Overview Dashboard

mac address table

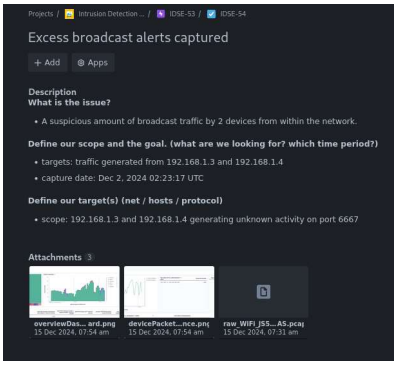
Top 15 values of eth_src.keyword	Top 15 values of eth_dst.keyword	(ipv4) IPv4 packet to broadcast dest address › Count of records	(ipv4) IPv4 datagram length > captured length › Count of records	(ipv4) IPv4 option set › Count of records	(stream) TCP SYN received › Count of records	(arp_spoof) unicast ARP request › Count of records	(ipv4) IPv4 packet from 'current net' source address › Count of records
84:E3:42:2F:0C:63	FF:FF:FF:FF:FF:FF	1,466	-	-	-	-	-
D4:A6:51:18:7E:94	FF:FF:FF:FF:FF:FF	1,459	-	-	-	-	-
A6:E8:2F:AD:C4:AA	FF:FF:FF:FF:FF:FF	4	-	-	-	-	-
E8:D8:19:88:50:03	FF:FF:FF:FF:FF:FF	4	-	-	-	-	4
EE:4F:01:83:9A:7E	FF:FF:FF:FF:FF:FF	3	-	-	-	-	3
80:2B:F9:9B:DA:BF	FF:FF:FF:FF:FF:FF	1	-	-	-	-	1
F4:CB:52:87:BD:D8	80:2B:F9:9B:DA:BF	1	126	-	-	11	-
A6:E8:2F:AD:C4:AA	01:00:5E:00:00:16	-	-	39	-	-	-

Example: Device Activity

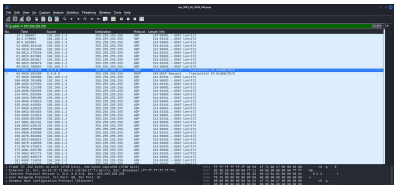


Analysis Process

Descriptive Analysis



Diagnostic Analysis



Predictive Analysis

Analysis Process
The analysis process helps to understand the bigger picture of an incident report. It can be broken into three key types:

1. **Descriptive Analysis**: What has happened?
2. **Diagnostic Analysis**: Drill into data to uncover previously unknown patterns.
3. **Predictive Analysis**: Provide a clear and concise summary of potential outcomes.

Baseline Network Traffic:
Establishing baseline network traffic is critical for detecting anomalies in alerts. Descriptive analysis provides a guide for determining the types of graphs to visualize.

What's Next?

- Automate the deployment of this pipeline using docker.
- Complete the analysis process on various network topologies and anomalies.
- Simulate attack patterns.
- Create a playbook for different attacks.