Project

Threat-Informed SOC Dashboard: Splunk-Based Detection with Snort and Sysmon, Aligned to MITRE ATT&CK and ATT&CK D3FEND

Executive Summary

This project presents the design and implementation of a threat-informed Security Operations Center (SOC) dashboard built on Splunk Enterprise. By integrating network-based detections from Snort and host-based telemetry from Sysmon, the dashboard correlates adversary activity with MITRE ATT&CK techniques and enriches each detection with corresponding MITRE D3FEND defensive countermeasures. The primary objective is to provide SOC analysts with a unified and actionable view of attacker behavior and recommended defensive responses within a single interface.

The virtual lab environment consists of a Kali Linux attacker machine, a Windows 10 victim endpoint running Sysmon and Splunk Universal Forwarder, and an Ubuntu server hosting Splunk Enterprise and Snort. A total of **26 custom Snort rules** were developed to detect reconnaissance, exploitation attempts, reverse shells, brute-force attacks, and other malicious behaviors. These detections were manually mapped to relevant ATT&CK techniques and visualized in interactive Splunk dashboards alongside applicable D3FEND techniques.

As a fully self-directed and independent project, this work demonstrates end-to-end SOC capabilities including telemetry ingestion, detection engineering, threat mapping, correlation, and visualization while showcasing the practical application of modern threat-informed defense principles. It lays a strong foundation for future enhancements, such as integration with SOAR platforms for automated response.

1. Introduction

Modern Security Operations Centers (SOCs) must rapidly interpret large volumes of alerts and determine appropriate responses in the face of evolving threats. Threat-informed defense addresses this challenge by leveraging adversary behavior models, such as the MITRE ATT&CK framework, to design detections and guide defensive actions. MITRE D3FEND complements ATT&CK by providing a structured catalog of defensive techniques that can be used to detect, mitigate, or counter these adversary behaviors.

Project Goal:
To design and implement a threat-informed SOC dashboard that maps detected attacks to corresponding defensive techniques using MITRE ATT&CK and D3FEND, enabling analysts to

view both adversary behavior and response guidance within a single Splunk interface.This project focuses on building the foundational detection and triage layer of a SOC, emphasizing:

- Visibility across host and network telemetry,
- Correlation of multi-source events, and
- Enhanced analyst decision support through unified threat-informed visualizations.

2. Project Roadmap and Methodology

The project was executed using a phased roadmap that mirrors the maturity progression of a real SOC environment:

1. Phase 1: Infrastructure Setup
2. Phase 2: Data Ingestion and Validation
3. Phase 3: Dashboard Development and Threat Mapping
4. Phase 4: Correlation and Analytics
5. Phase 5: Attack Simulation and Testing

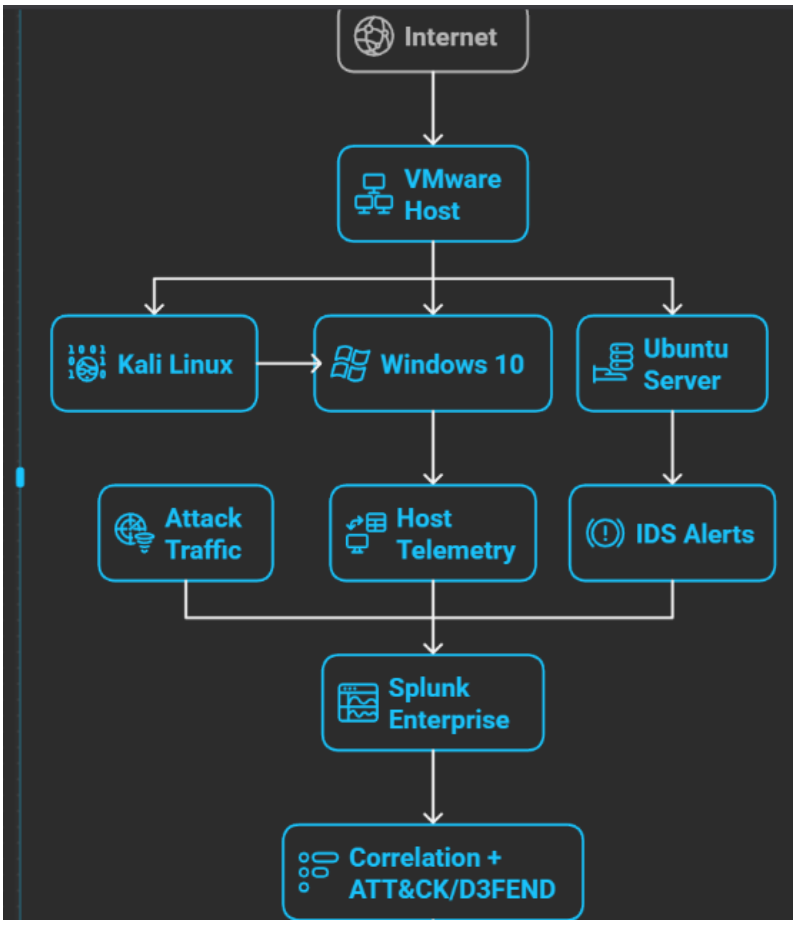
Each phase builds upon the previous one, ensuring a structured and engineering-driven development process

3. Lab Architecture and Environment

3.1 Systems Overview

System	OS	Role	IP Address
Kali Linux	Kali	Attacker machine	192.168.160.133
Windows 10	Windows 10	Victim endpoint (Sysmon + Splunk Universal Forwarder)	192.168.160.131
Ubuntu Server	Ubuntu	SOC server (Splunk Enterprise + Snort IDS)	192.168.160.132

All IP addresses shown are private addresses assigned within an isolated virtual lab network.



3.2 Network Configuration

Each VM is configured with two network adapters:

- Host-Only Adapter: Provides an isolated internal network for attack simulation and telemetry exchange between Kali, Windows, and Ubuntu.
- NAT Adapter: Provides controlled Internet access for system updates, tool installation, and package downloads.

This dual-adapter design ensures the lab remains isolated while still allowing necessary external connectivity.

3.3 Architecture Description

The lab environment simulates a small enterprise SOC deployment. Kali Linux generates adversary activity targeting the Windows 10 endpoint. Host-based telemetry is collected using

Sysmon and forwarded via Splunk Universal Forwarder to Splunk Enterprise running on the Ubuntu SOC server. Network traffic between the attacker and victim is monitored by Snort IDS on Ubuntu, which generates alerts for suspicious activity. Both host logs and IDS alerts are ingested into Splunk for centralized analysis, correlation, and visualization. Splunk serves as the core SIEM platform, where detections are mapped to MITRE ATT&CK techniques and enriched with relevant MITRE D3FEND defensive guidance, enabling a threat-informed SOC workflow.

3.4 Data Flow

1. Kali Linux launches attacks against Windows 10.
2. Sysmon records host activity on Windows.
3. Splunk Universal Forwarder sends Sysmon logs to Splunk Enterprise.
4. Snort monitors network traffic and generates IDS alerts.
5. Snort alerts are ingested into Splunk Enterprise.
6. Splunk correlates events and maps them to ATT&CK and D3FEND.
7. Results are visualized in SOC dashboards.

3.5. Tools and Technologies

- Splunk Enterprise :SIEM platform for log ingestion, search, correlation, and visualization.
- Splunk Universal Forwarder :Lightweight agent to forward Windows logs.
- Sysmon : Host-based monitoring for detailed Windows telemetry.
- Snort IDS : Network intrusion detection system with custom rules.
- Kali Linux :Offensive security platform to simulate attacks.
- MITRE ATT&CK :Adversary behavior framework for detection mapping.
- MITRE D3FEND :Defensive countermeasure framework for mitigation guidance.
- VMware Workstation : Virtualization platform for lab deployment.

4. Phase 1 – Infrastructure Setup

4.1 Virtualization Setup

VMware Workstation was used to deploy three VMs with appropriate resource allocation. All systems were configured to boot automatically and maintain consistent IP addressing within the host-only network.

4.2 Sysmon on Windows 10 – Host Telemetry Layer

Sysmon was installed on the Windows endpoint to provide fine-grained visibility into system activity. Unlike default Windows logs, Sysmon captures:

- Process creation with command lines,
- Network connections initiated by processes,
- File creation and modification,
- Registry changes.

This level of detail is critical in SOC investigations to reconstruct attacker behavior after an alert is raised.

A custom Sysmon configuration file was used to focus on events most relevant to detection and reduce noise.

Why Sysmon:
It transforms the Windows endpoint into a rich sensor capable of exposing attacker techniques such as execution, persistence, and lateral movement.

Command used :

sysmon64.exe -accepteula -i sysmonconfig.xml

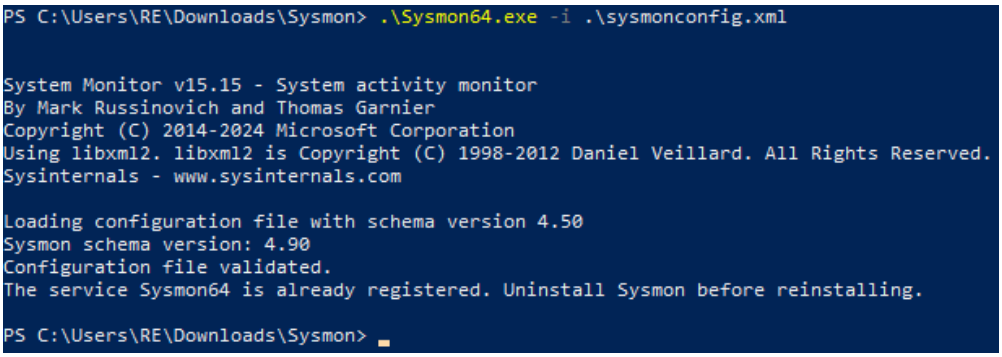


Fig 1 : confirmation of Sysmon config

4.3 Splunk Universal Forwarder – Log Shipping Mechanism

The Splunk Universal Forwarder (UF) was installed on Windows 10 to securely transmit Sysmon logs to the Splunk Enterprise server. In SOC architectures, forwarders serve as lightweight agents that:

- Collect local logs,
- Compress and securely send them, and
- Minimize performance impact on endpoints.

The forwarder was configured to:

- Send data to 192.168.153.132:9997,
- Monitor the Sysmon event log channel.

This ensures that every Sysmon event generated on Windows becomes searchable in Splunk.

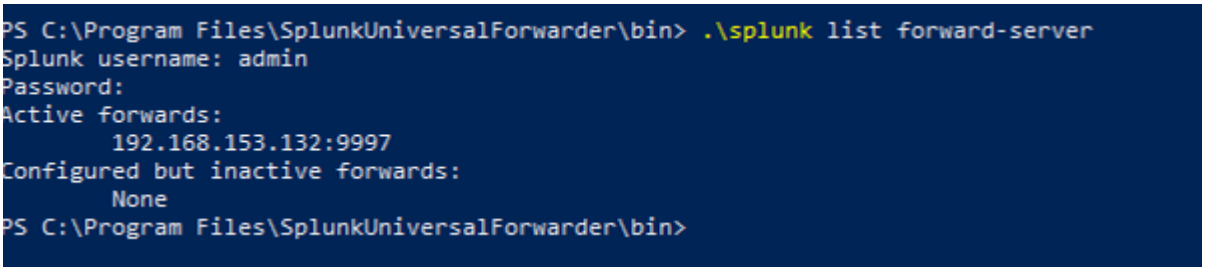


Fig 2 : Splunk Universal Forwarder on Windows 10 showing active forwarding configuration to the Splunk Enterprise server.

4.4 Splunk Enterprise on Ubuntu

Splunk Enterprise was deployed on the Ubuntu server to act as the central collection and analysis platform. In a SOC, the SIEM is responsible for:

- Ingesting logs from diverse sources,
- Indexing data for fast search,
- Correlating events across systems, and
- Powering dashboards for analysts.

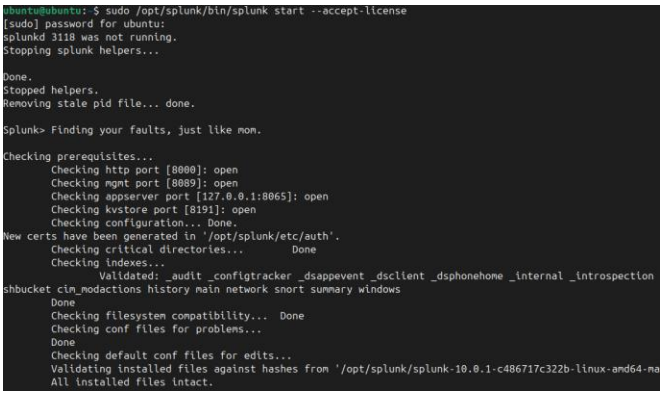
After installation, Splunk was configured to:

- Run as a persistent service,
- Listen on TCP port 9997 for incoming forwarder data, and
- Provide access through Splunk Web (port 8000).

By enabling the receiving port, Splunk becomes capable of acting as the log aggregation point for endpoint telemetry and IDS alerts.

Command Used :

sudo /opt/splunk/bin/splunk start --accept-license



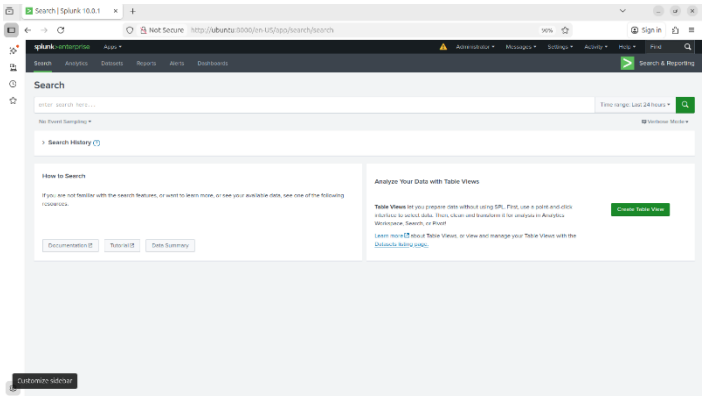


Fig 3: Splunk Enterprise started successfully on Ubuntu and accessible via Splunk Web.

4.5 Snort IDS on Ubuntu – Network Detection Layer

Snort was installed on the same Ubuntu system to perform network intrusion detection. In real SOCs, IDS/IPS tools analyze packet-level traffic to identify:

- Scans and reconnaissance,
- Exploitation attempts,
- Malware callbacks

Snort was configured to monitor the host-only interface, ensuring visibility into traffic between Kali and Windows. A set of 26 custom Snort rules was developed to detect behaviors such as:

- Port scanning,
- SMB brute force,
- Reverse shells,
- Suspicious payload patterns.

Custom rules allow detection logic to be aligned with ATT&CK techniques rather than relying only on generic signatures.

Command Used :

sudo snort -i ens33 -c /etc/snort/snort.lua -A fast -l /var/log/snort

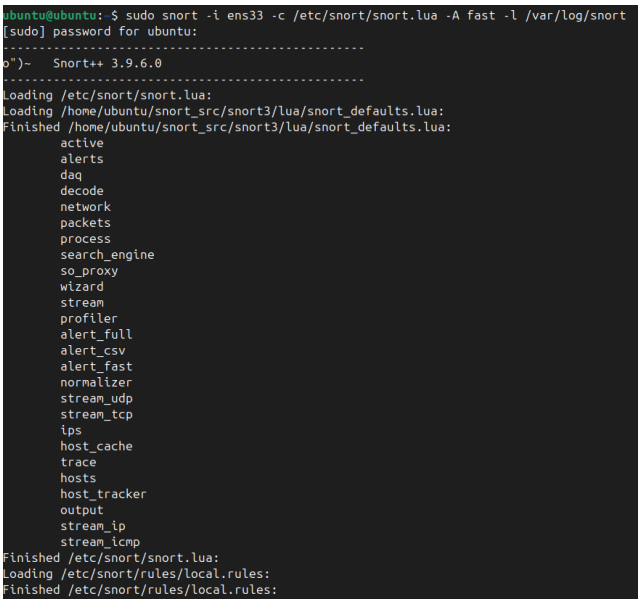


Fig 4 : Snort IDS successfully initialized on the Ubuntu SOC server

5 Phase 2 – Data Ingestion and Validation

After deploying Sysmon on the Windows endpoint, configuring the Splunk Universal Forwarder, and enabling Snort on the Ubuntu SOC server, data ingestion into Splunk Enterprise was validated to ensure end-to-end telemetry flow.

To verify network-based detections, the following search was executed:

index=snort

The results confirmed that Snort alerts generated on the Ubuntu server were successfully indexed in Splunk, with events sourced from /var/log/snort/alert_fast.txt and labeled with the appropriate source type. This validates that the IDS alerts are being collected and made available for correlation and analysis.

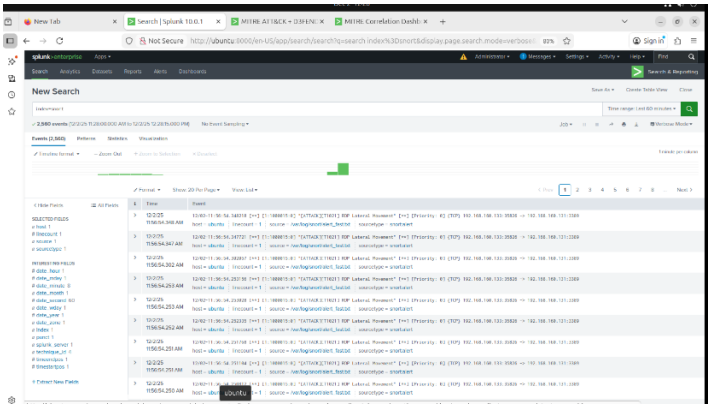


Fig 5 : Snort IDS alerts successfully indexed in Splunk, confirming ingestion of network-based detection data.

To verify host-based telemetry from the Windows endpoint, the following search was executed:

Index=windows

The results showed Sysmon operational events forwarded by the Splunk Universal Forwarder, including detailed process, network, and system activity from the Windows 10 host. This confirms that endpoint telemetry is being reliably transmitted to the SIEM

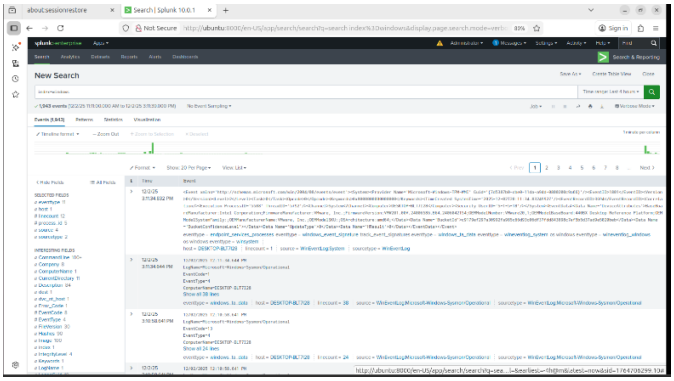


Fig 6 : Windows Sysmon events indexed in Splunk, confirming successful ingestion of host-based telemetry via Universal Forwarder.

Together, these results demonstrate that both host and network sensors are actively streaming data into Splunk Enterprise, establishing the foundation for multi-source correlation and threat-informed analysis.

6. Phase 3 – Dashboard Development and Threat Mapping

Once host-based telemetry from Sysmon and network-based alerts from Snort were successfully ingested into Splunk Enterprise, the next step was to correlate these events and enrich them with adversary and defensive context using the MITRE ATT&CK and MITRE D3FEND frameworks.

In a real SOC, raw alerts alone are insufficient for effective triage. Analysts require contextual information that explains:

- what the attacker is attempting to achieve, and
- which defensive actions are relevant.

This project operationalizes that concept by mapping detections to ATT&CK techniques and associating them with D3FEND defensive countermeasures.

6.1 MITRE ATT&CK Lookup Integration

To enable threat-informed detection, a MITRE ATT&CK lookup table was integrated into Splunk to enrich raw IDS alerts with adversary behavior context. This lookup allows each detection to be mapped to a standardized ATT&CK technique and tactic, transforming low-level events into meaningful threat intelligence.

A CSV file named mitre_attack_lookup.csv was created containing mappings between custom Snort rule identifiers (SIDs) and corresponding MITRE ATT&CK metadata. Each entry in the lookup includes the following fields:

- snort_sid – Snort rule identifier
- technique_id – MITRE ATT&CK technique ID (e.g., T1046)
- technique_name – ATT&CK technique name
- tactic_name – ATT&CK tactic category (e.g., Reconnaissance, Execution)

The lookup file was uploaded into Splunk using the Settings → Lookups → Lookup table files interface and saved with the destination name mitre_attack_lookup, making it accessible for SPL-based enrichment.

To validate successful integration, the following search was executed:

| inputlookup mitre_attack_lookup

| table technique_id, technique_name, tactic_name, snort_sid

| head 10

This query confirms that the lookup table is properly loaded and that the ATT&CK mappings are available for use within Splunk. Figure X shows the lookup configuration and validation results

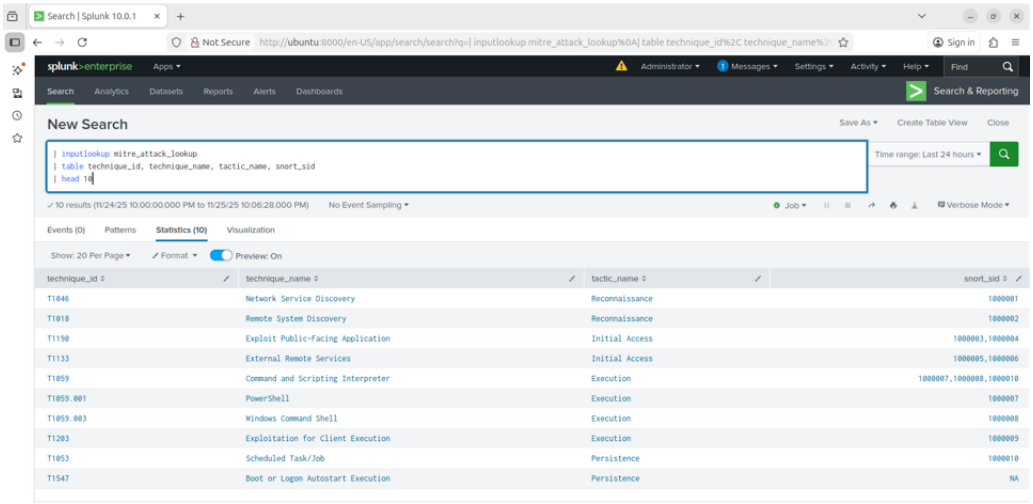


Fig 7 : Shows the successful output of the lookup validation query

6.2 MITRE D3FEND Lookup Integration

To provide defensive context alongside adversary behavior, a MITRE D3FEND lookup integration was implemented in Splunk. While the MITRE ATT&CK framework describes attacker tactics and techniques, MITRE D3FEND catalogs defensive techniques that can detect, harden, or mitigate those behaviors. Integrating D3FEND enables the SOC dashboard to present both attack and defense perspectives within a single interface.

A CSV lookup file named d3fend_mapping.csv was created to map ATT&CK technique IDs to corresponding D3FEND defensive techniques. Each entry in the lookup includes the following fields:

- technique_id – MITRE ATT&CK technique ID
- d3fend_technique – Defensive technique name
- d3fend_tactic – Defensive category (e.g., Detect, Isolate, Harden)
- defense_description – Description of the defensive action

The lookup file was uploaded and configured in Splunk using **Settings → Lookups → Lookup table files**, making it available for SPL-based enrichment.

After Snort alerts were enriched with ATT&CK context, the D3FEND lookup was applied using the ATT&CK technique ID as the correlation key. The following SPL query was used to enrich detections with defensive techniques:

index=snort sourcetype="snort:alert"

| lookup mitre_attack_lookup technique_id OUTPUT technique_name, tactic_name, tactic_order

| lookup d3fend_mapping technique_id OUTPUT d3fend_technique, d3fend_tactic, defense_description

| stats count by technique_id, technique_name, d3fend_technique, d3fend_tactic

| sort tactic_order, -count

This query enriches each detection with both adversary behavior and recommended defensive techniques, and aggregates results to highlight the most frequent attack–defense relationships

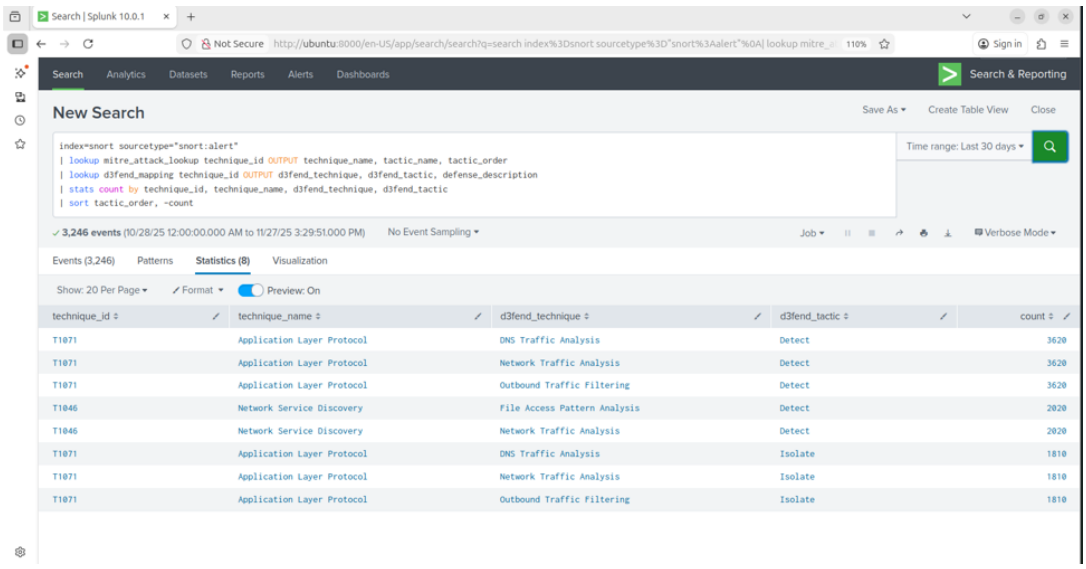


Fig 8 : Shows the successful output of the lookup validation query

6.3 Threat-Informed Correlation Dashboard

A threat-informed SOC dashboard titled “Threat Intelligence – MITRE ATT&CK Overview” was developed in Splunk to visualize correlated detections enriched with both MITRE ATT&CK and MITRE D3FEND context. This dashboard serves as the primary interface for analysts to understand attacker behavior and corresponding defensive techniques within a single view.

The dashboard is powered by enriched Snort IDS alert data and presents real-time visibility into detected adversary activity mapped to standardized threat models. It aggregates detections using SPL searches that apply both ATT&CK and D3FEND lookups and summarizes results for SOC-level analysis.

The core SPL driving the dashboard panels is:

index=snort sourcetype="snort:alert"

| lookup mitre_attack_lookup technique_id OUTPUT technique_name, tactic_name, tactic_order

| lookup d3fend_mapping technique_id OUTPUT d3fend_technique, d3fend_tactic, defense_description

| stats count by technique_id, technique_name, tactic_name, d3fend_technique, d3fend_tactic

| sort tactic_order, -count

Dashboard Components

The dashboard includes the following visual elements:

- ATT&CK Tactic Distribution: A pie chart showing the proportion of detected events across ATT&CK tactics such as Reconnaissance, Execution, Persistence, and Impact.
- Top ATT&CK Techniques: A bar chart highlighting the most frequently observed techniques (e.g., Network Service Discovery, Application Layer Protocol).
- D3FEND Defensive Actions: A visualization summarizing defensive techniques categorized by D3FEND tactics such as Detect, Isolate, and Harden.
- Key Metrics Panels: Single-value panels displaying the total number of detections and unique ATT&CK techniques observed within the selected time range.
- Enriched Event Table: A table listing ATT&CK technique IDs, technique names, mapped D3FEND techniques, and event counts.

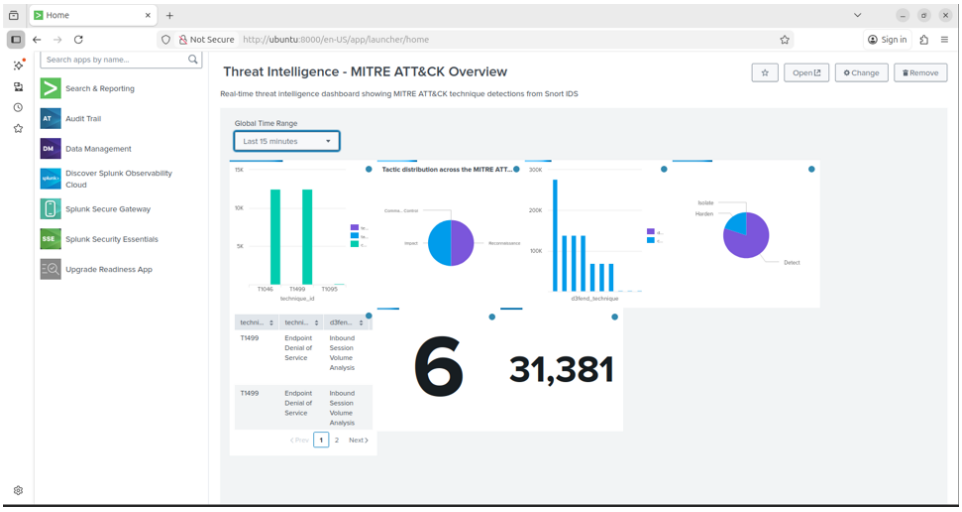


Fig 9 : shows the Threat Intelligence – MITRE ATT&CK Overview dashboard created as part of this project.

7. Phase 4 – Correlation and Detection Engineering

While Phase 3 focused on enriching detections and visualizing threat intelligence using MITRE ATT&CK and D3FEND, Phase 4 emphasizes detection engineering and correlation logic. The objective of this phase is to transform raw IDS and endpoint events into actionable security detections that represent meaningful adversary behaviors.

In a real SOC, individual alerts rarely provide sufficient context. Correlation across time, hosts, and data sources is required to identify attack patterns such as brute-force attempts, lateral movement, and malicious execution. This phase implements SPL-based analytics in Splunk to group related events, reduce noise, and surface high-confidence detections aligned to ATT&CK techniques.

7.1 SSH Brute Force Correlation – MITRE ATT&CK T1110

One of the primary detections implemented in this project is SSH brute-force activity against the Ubuntu server, mapped to MITRE ATT&CK technique T1110 (Brute Force) under the Credential Access tactic.

Snort IDS generates alerts for repeated SSH login attempts. However, individual alerts do not indicate whether the activity represents a true attack. To address this, correlation logic was developed in Splunk to aggregate multiple failed attempts from the same attacker against a target host and port within a time window.

The following SPL query was used to correlate SSH brute-force attempts:

The following SPL query was used to correlate SSH brute-force attempts:

```
index=snort sourcetype="snort:alert"
| rex field=_raw "(?<attacker>\d+\.\d+\.\d+\.\d+):\d+\s+->\s+(?<target>\d+\.\d+\.\d+\.\d+):(?<port>\d+)"
| lookup mitre_attack_lookup technique_id OUTPUT technique_name, tactic_name
| stats count as attempts, min(_time) as first_attempt, max(_time) as last_attempt
    by attacker, target, port, technique_name
| eval duration=last_attempt-first_attempt
| where attempts > 10
| sort -attempts
```

This correlation logic:

- Extracts attacker and target IP addresses using regex.

- Aggregates events by source, destination, and technique.
- Calculates total attempts and attack duration.
- Filters out low-frequency activity to reduce false positives.

The output highlights sustained brute-force behavior, providing SOC analysts with a high-confidence detection rather than isolated alerts.

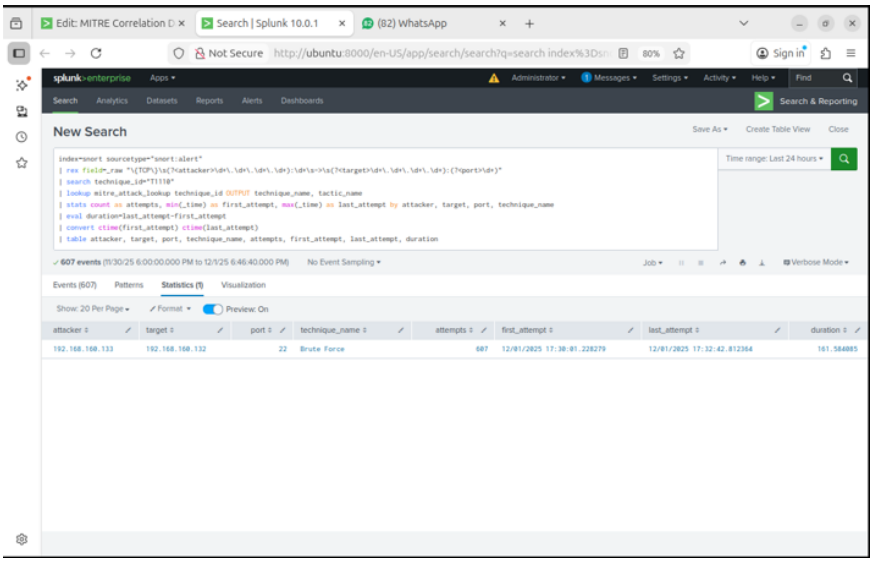


Fig 10 : shows the correlated SSH brute-force detection results mapped to ATT&CK T1110 in Splunk

7.2 MITRE Correlation Dashboard

To operationalize the correlation logic, a dedicated dashboard titled **“MITRE Correlation Dashboard”** was created in Splunk. This dashboard presents correlated detections aligned to ATT&CK techniques, enabling analysts to quickly assess active attack patterns.

The dashboard displays:

- Attacker IP address
- Target system
- Target port
- ATT&CK technique name
- Number of attempts
- First and last observed timestamps
- Attack duration

This structured view transforms raw IDS alerts into SOC-ready detections that reflect attacker behavior over time.

The dashboard enables analysts to:

- Prioritize high-volume or long-duration attacks.
- Identify repeated targeting of critical assets.
- Pivot into raw events for deeper investigation.

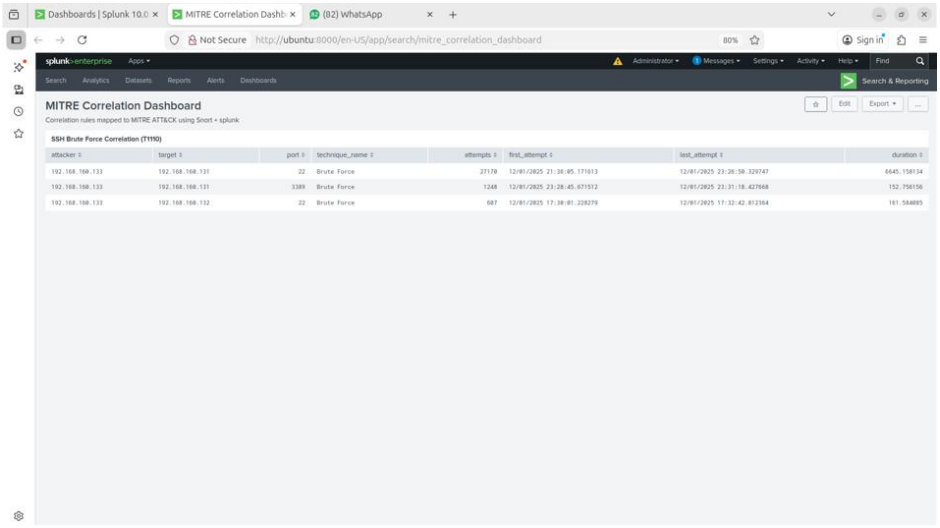


Fig 11 : shows the MITRE Correlation Dashboard highlighting SSH brute-force and RDP activity detected during attack simulation from the Kali Linux attacker.

7.3 Host Telemetry Validation Using Sysmon (Process Execution Events)

In addition to network-based detections from Snort, host-level visibility was validated using Sysmon logs collected from the Windows 10 endpoint. Sysmon provides detailed telemetry on process creation, command-line execution, and other endpoint activities, which are critical for endpoint detection and response in a SOC.

To verify that Sysmon events were being successfully generated and ingested into Splunk, benign process executions were performed on the Windows endpoint, including launching **notepad.exe** and **calc.exe**. These actions simulate typical user activity and confirm that process creation events are captured correctly.

The following SPL queries were used to validate Sysmon Event ID 1 (Process Create):

index=windows EventCode=1 Image="*notepad.exe" earliest=-30m

| table _time Image CommandLine User

index=windows EventCode=1 Image="*calc.exe" earliest=-30m

| table _time Image CommandLine User

These searches return the timestamp, executable path, command-line arguments, and user context for each process execution. The successful results confirm that:

- Sysmon is correctly logging process creation events,
- The Splunk Universal Forwarder is forwarding Windows event logs, and
- Splunk Enterprise is indexing and making the data available for analysis.

Such telemetry forms the foundation for detecting attacker behaviors such as command execution, scripting abuse, or living-off-the-land techniques in later detection use cases.

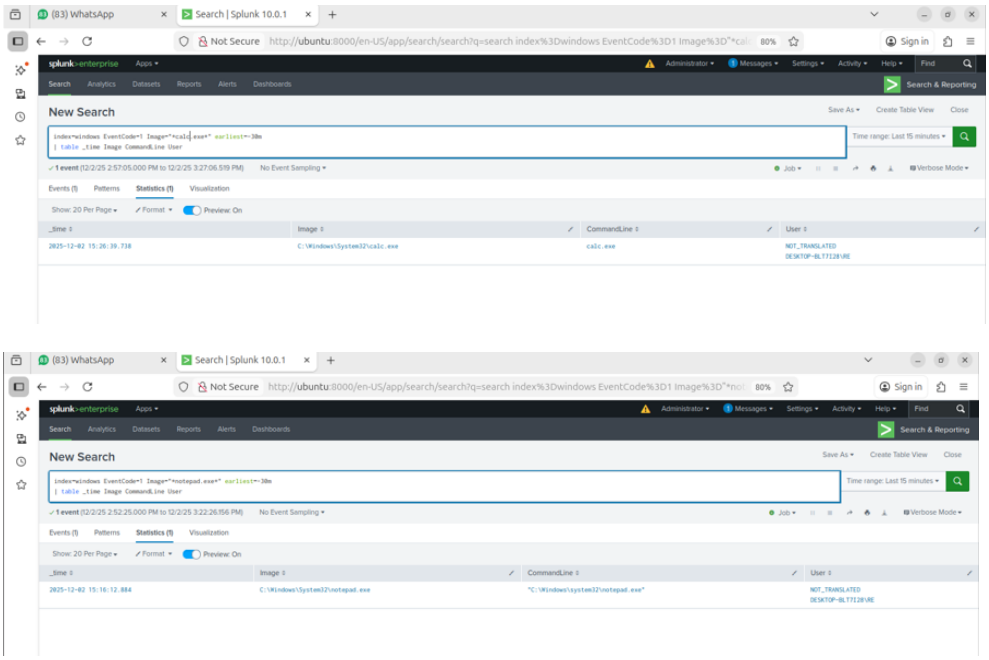


Fig 12: shows the Splunk search results for Notepad execution, and the corresponding results for

8. Phase 5 – Attack Simulation and Detection Validation

This phase validates the end-to-end SOC pipeline by generating realistic adversary activity from the Kali Linux attacker VM and verifying that Snort IDS, Sysmon, and Splunk Enterprise successfully detect, ingest, enrich, and correlate the events using the MITRE ATT&CK and D3FEND frameworks.

The objective is to demonstrate that the deployed lab environment can:

- simulate common attacker techniques,
- detect both network- and host-based behaviors, and
- present actionable, threat-informed intelligence to analysts.

8.1 Attack Simulation Setup

To emulate real-world adversary behavior, multiple attack techniques were executed from the Kali Linux VM targeting the Windows 10 and Ubuntu systems within the virtual SOC network.

8.1.1 Environment and Roles

- Attacker: Kali Linux (192.168.160.133)
- Targets:
    - Windows 10 endpoint (Sysmon enabled)
    - Ubuntu Server hosting Splunk and Snort (192.168.160.132)
- Monitoring:
    - Snort IDS monitoring network traffic
    - Sysmon collecting host telemetry on Windows
    - Splunk Enterprise aggregating and enriching logs

All VMs were connected using a Host-Only adapter for internal traffic and a NAT adapter for Internet access.

8.1.2 Reconnaissance – Network Scanning (ATT&CK T1046)

An Nmap scan was launched from Kali to discover open ports and services on the target:

nmap -A 192.168.160.132

This scan identified exposed services including:

- TCP 80 (HTTP – Apache),
- TCP 3389 (RDP),
- TCP 8000/8089 (Splunk web and management).

Purpose:
Simulate attacker reconnaissance to enumerate services and potential attack vectors.

Expected Detection:
Snort rule mapped to **T1046 – Network Service Discovery**.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A 192.168.160.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-26 00:05 EST
Nmap scan report for 192.168.160.132
Host is up (0.00046s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
3389/tcp open  ms-wbt-server Microsoft Terminal Service
8000/tcp open  http          Splunkd httpd
|_http-server-header: Splunkd
| http-robots.txt: 1 disallowed entry
|_/
| http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_Requested resource was http://192.168.160.132:8000/en-US/account/login?return_to=%2Fen-US%2F
8089/tcp open  ssl/http      Splunkd httpd
| http-robots.txt: 1 disallowed entry
|_/
| ssl-cert: Subject: commonName=SplunkServerDefaultCert/organizationName=SplunkUser
| Not valid before: 2025-10-27T17:12:18
|_Not valid after:  2028-10-26T17:12:18
|_http-title: splunkd
MAC Address: 00:0C:29:C5:A5:A0 (VMware)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kern
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

TRACEROUTE
HOP RTT     ADDRESS
1   0.46 ms 192.168.160.132

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.96 seconds
```

Fig 13 : An Nmap scan was launched from Kali to discover open ports and services on the target

Snort IDS, deployed on the Ubuntu SOC server, was configured with custom and community rules to detect port scanning behavior. When the scan was executed, Snort generated alerts labeled **"Port Scan Detected"**, which were explicitly mapped to **MITRE ATT&CK technique T1046 (Network Service Discovery)** within the alert message.

These alerts indicate repeated connection attempts from the attacker IP (**192.168.160.133**) targeting multiple ports on the victim host (**192.168.160.132**), which is characteristic of service enumeration during the reconnaissance phase.

The following alert format was observed in /var/log/snort/alert_fast.txt:

```
12/26-00:06:01.572679 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46968 -> 192.168.160.132:80
12/26-00:06:01.572840 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
12/26-00:06:01.573191 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
12/26-00:06:01.573463 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32924 -> 192.168.160.132:8000
12/26-00:06:01.577523 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32936 -> 192.168.160.132:8000
12/26-00:06:01.580585 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46974 -> 192.168.160.132:80
12/26-00:06:01.581871 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46980 -> 192.168.160.132:80
12/26-00:06:01.582025 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:46998 -> 192.168.160.132:80
12/26-00:06:01.582425 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46998 -> 192.168.160.132:80
12/26-00:06:01.582426 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:47012 -> 192.168.160.132:80
12/26-00:06:01.582427 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32944 -> 192.168.160.132:8000
12/26-00:06:01.582673 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47012 -> 192.168.160.132:80
12/26-00:06:01.582950 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32954 -> 192.168.160.132:8000
12/26-00:06:01.583086 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.583223 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.586777 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
12/26-00:06:01.601042 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
12/26-00:06:01.609028 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46952 -> 192.168.160.132:80
12/26-00:06:01.638059 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:48564 -> 192.168.160.132:8089
12/26-00:06:01.639000 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:48566 -> 192.168.160.132:8089
12/26-00:06:01.639006 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32956 -> 192.168.160.132:8000
12/26-00:06:01.639163 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:47028 -> 192.168.160.132:80
12/26-00:06:01.639345 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47028 -> 192.168.160.132:80
12/26-00:06:01.639453 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:48568 -> 192.168.160.132:8089
12/26-00:06:01.640991 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46998 -> 192.168.160.132:80
12/26-00:06:01.640992 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47012 -> 192.168.160.132:80
12/26-00:06:01.641488 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.645006 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
12/26-00:06:01.659626 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.659628 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.659943 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47022 -> 192.168.160.132:80
12/26-00:06:01.662551 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46998 -> 192.168.160.132:80
12/26-00:06:01.665603 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:47012 -> 192.168.160.132:80
12/26-00:06:01.691735 [**] [1:1000001:0] "[ATTACK][T1046] Port Scan Detected" [**] [Priority: 0] {TCP} 192.168.160.133:32960 -> 192.168.160.132:8000
12/26-00:06:01.694025 [**] [1:1000017:0] "[ATTACK][T1071] HTTP C2 Beaconing" [**] [Priority: 0] {TCP} 192.168.160.133:46982 -> 192.168.160.132:80
```

Fig 14 : Snort alerts showing "Port Scan Detected" mapped to MITRE ATT&CK T1046 (Network Service Discovery)

All Snort alerts were forwarded to Splunk Enterprise and indexed under the snort index, where they were further enriched using the MITRE ATT&CK lookup table. This enrichment enabled the SOC dashboard to classify the activity under the **Reconnaissance** tactic and visualize it as part of the threat-informed analysis.

8.1.3 Brute Force Attempts – Remote Services (ATT&CK T1110, T1021)

Brute force activity was simulated using Hydra to attempt authentication attacks against remote services. For example:

hydra -l ubuntu -P /usr/share/wordlists/rockyou.txt RDP://192.168.160.132 -t 4 -V

Although brute force was attempted, the environment also generated repeated connections to TCP **3389 (RDP)** during scanning and probing activity, which triggered Snort rules for:

- **T1021 – Remote Services (RDP)**
- **T1110 – Brute Force**

Purpose:
Simulate credential access attempts against exposed services.

Expected Detection:
Snort alerts for brute force and remote service access.

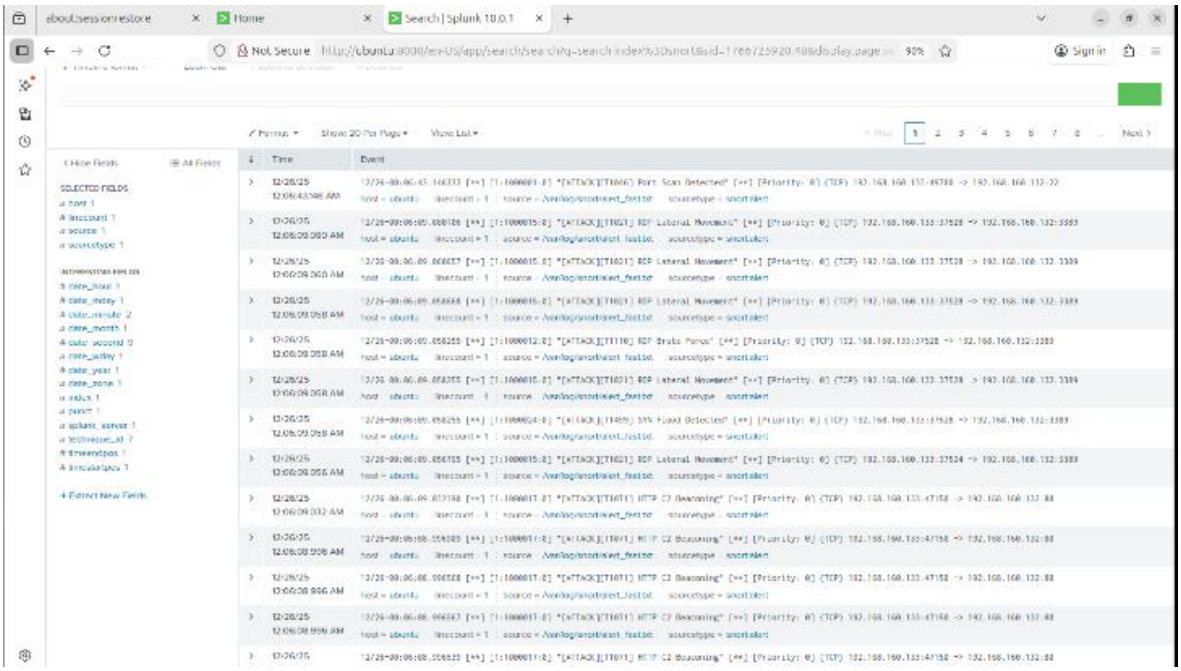


Fig 15 : Snort alerts were forwarded to Splunk Enterprise and indexed under the snort index

8.2 Detection and Validation in Splunk

Following the execution of attack scenarios from the Kali Linux attacker, the SOC pipeline was validated end-to-end by confirming that Snort IDS and Sysmon generated alerts, that these events were ingested into Splunk Enterprise, and that detections were enriched with MITRE ATT&CK and MITRE D3FEND context.

This phase demonstrates that the lab environment successfully transitions from raw telemetry to actionable, threat-informed intelligence.

8.2.1 Validation of Reconnaissance Detection (ATT&CK T1046)

The Nmap scan launched against the Ubuntu SOC server triggered multiple Snort alerts labeled “Port Scan Detected”, mapped to MITRE ATT&CK T1046 – Network Service Discovery.

In Splunk, these alerts were validated using the following query:

index=snort sourcetype="snort:alert" T1046

The results confirmed:

- Source IP: Kali attacker (192.168.160.133)
- Destination IP: Ubuntu server (192.168.160.132)
- Multiple destination ports scanned
- Enrichment with ATT&CK technique and Reconnaissance tactic

These detections appeared in both the raw search view and the **Threat Intelligence – MITRE ATT&CK Overview dashboard**, contributing to the Reconnaissance tactic distribution.

Outcome:
The SOC successfully detected reconnaissance activity and classified it under ATT&CK T1046 with corresponding D3FEND recommendations such as Network Traffic Analysis and File Access Pattern Analysis

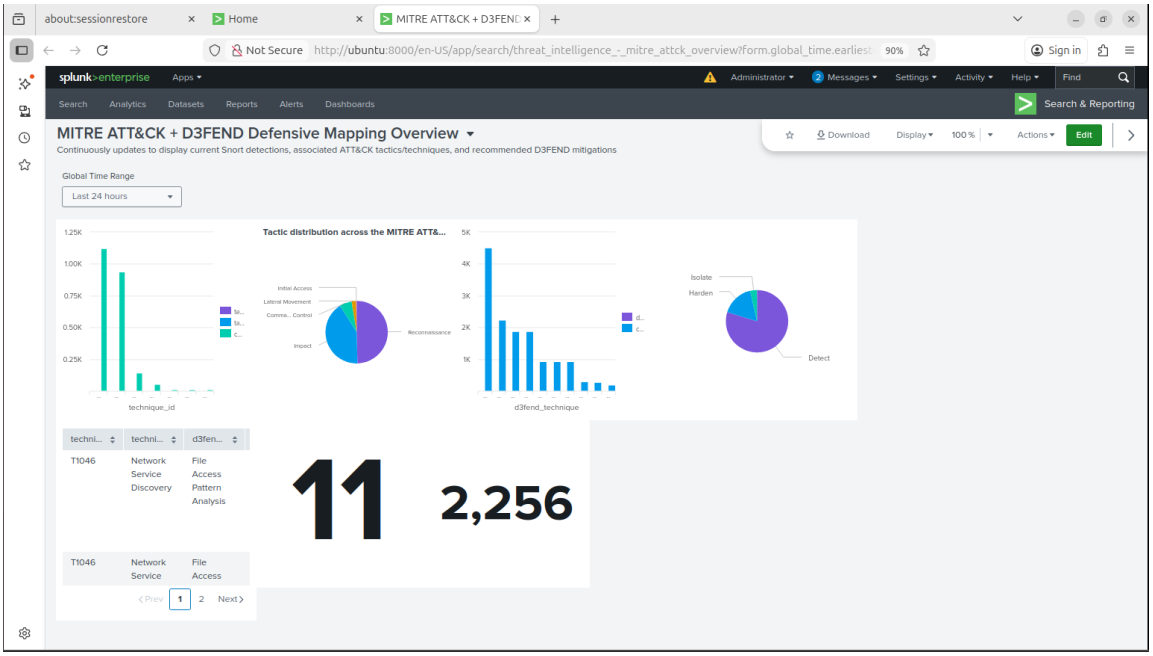


Fig 16 : Threat-Informed Dashboard showing Snort detections mapped to MITRE ATT&CK techniques and associated D3FEND defensive actions for reconnaissance activity (T1046)

8.2.2 Validation of Brute Force and Remote Services (ATT&CK T1110, T1021)

Brute-force style activity and repeated connections to RDP (TCP 3389) generated Snort alerts mapped to:

- **T1110 – Brute Force**
- **T1021 – Remote Services**

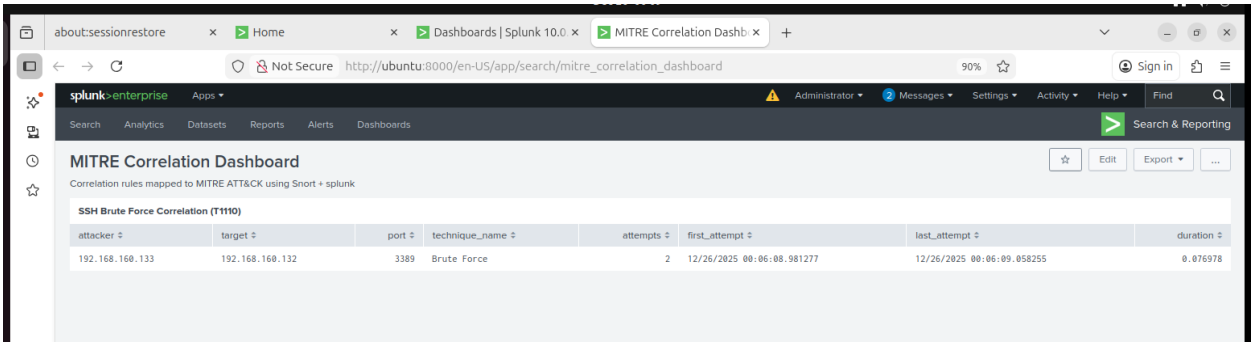


Fig 17: Correlation logic successfully grouped repeated attempts into a single high-confidence detection, mapped to ATT&CK T1110 and enriched with D3FEND techniques such as Inbound Session Volume Analysis and Isolate.

technique_id	technique_name	d3fend_technique	defense_description	count
T1046	Network Service Discovery	File Access Pattern Analysis	Analyze file access patterns to detect service enumeration	1123
T1046	Network Service Discovery	File Access Pattern Analysis	Monitor network traffic for port scanning and service enumeration patterns	1123
T1046	Network Service Discovery	Network Traffic Analysis	Analyze file access patterns to detect service enumeration	1123
T1046	Network Service Discovery	Network Traffic Analysis	Monitor network traffic for port scanning and service enumeration patterns	1123
T1499	Endpoint Denial of Service	Inbound Session Volume Analysis	Analyze traffic patterns for DoS indicators	937
T1499	Endpoint Denial of Service	Inbound Session Volume Analysis	Detect abnormal traffic volumes indicating DoS attacks	937
T1499	Endpoint Denial of Service	Inbound Session Volume Analysis	Implement rate limiting to mitigate DoS attacks	937
T1499	Endpoint Denial of Service	Network Traffic Analysis	Analyze traffic patterns for DoS indicators	937
T1499	Endpoint Denial of Service	Network Traffic Analysis	Detect abnormal traffic volumes indicating DoS attacks	937
T1499	Endpoint Denial of Service	Network Traffic Analysis	Implement rate limiting to mitigate DoS attacks	937

< Prev 1 2 Next >

Fig 18 : D3FEND techniques such as Inbound Session Volume Analysis and Isolate.

9. Results and Key Findings

This section presents the outcomes of implementing the threat-informed SOC dashboard and evaluates how effectively the lab environment detected, correlated, and contextualized simulated attacks using MITRE ATT&CK and D3FEND.

9.1 Detection Coverage and Visibility

The deployed SOC environment successfully ingested and correlated host-based telemetry from Sysmon and network-based alerts from Snort into Splunk Enterprise. The dashboard provided real-time visibility into multiple attack techniques, including:

- **T1046 – Network Service Discovery** (port scanning),
- **T1110 – Brute Force** (SSH/RDP login attempts),
- **T1071 – Application Layer Protocol** (HTTP-based beaconing),
- **T1499 – Endpoint Denial of Service** (traffic flooding patterns).

Across the evaluation window, the dashboard displayed:

- Total detections in the range of thousands of events,
- Multiple unique ATT&CK techniques observed,
- Clear dominance of Reconnaissance and Execution tactics, reflecting early-stage attack activity.

This demonstrates that the SOC stack achieved broad visibility across both host and network layers and that the enrichment process successfully transformed raw alerts into structured threat intelligence.

9.2 Threat–Defense Correlation Using ATT&CK and D3FEND

By enriching Snort detections with ATT&CK and D3FEND lookups, each alert was mapped not only to adversary behavior but also to corresponding defensive actions. For example:

- **T1046 – Network Service Discovery** was mapped to:
    - D3FEND: Network Traffic Analysis and
    - File Access Pattern Analysis (Detect).
- **T1110 – Brute Force** was mapped to:
    - Inbound Session Volume Analysis (Detect/Isolate).

This dual mapping enabled analysts to immediately see:

- What the attacker is doing, and
- How the defender should respond.

The dashboard validated the feasibility of embedding defensive guidance directly into SOC workflows, improving analyst decision support and reducing time to response.

9.3 Correlation Effectiveness

Correlation searches grouped multiple low-level alerts into higher-level attack patterns. For example:

- Repeated SSH connection attempts from Kali to Ubuntu were correlated into a single **Brute Force (T1110)** event with:
    - Attacker IP,
    - Target IP,
    - Attempt count,
    - First and last attempt timestamps,
    - Attack duration.

This reduced alert fatigue and demonstrated how correlation can convert noisy IDS data into actionable incidents.

9.4 False Positives and Detection Tuning Observations

During analysis, a notable false positive was observed:

Snort generated **T1046 – Network Service Discovery** alerts for traffic originating from the Windows endpoint toward the Ubuntu Splunk server. Investigation revealed that this traffic was produced by the **Splunk Universal Forwarder** while establishing connections to forward logs.

Although the traffic pattern matched port scan heuristics, it was in fact legitimate SOC infrastructure communication.

This highlights an important SOC lesson:

Without contextual awareness, benign operational traffic can resemble adversary behavior.

In a production SOC, this would be addressed through:

- Asset role tagging (e.g., marking Splunk forwarders),
- Rule tuning or whitelisting trusted IPs,
- Context-aware correlation.

This project intentionally retained the alert to demonstrate the importance of analyst validation and detection tuning in threat-informed defense.

9.5 Summary of Results

Overall, the project demonstrated that:

- Multi-source telemetry can be effectively centralized in Splunk,
- ATT&CK enrichment improves alert interpretability,
- D3FEND mapping provides immediate defensive guidance,
- Correlation reduces noise and highlights meaningful attack patterns,
- Contextual analysis is required to handle false positives.

Overall, the SOC dashboard successfully met its objective of providing unified visibility into adversary behavior and defensive guidance.

9.6 Lessons Learned

This project provided several important insights into practical SOC operations and threat-informed defense:

1. **Context is essential for accurate detection.**
   Raw IDS alerts alone are insufficient to determine malicious intent. The false positive where Splunk Universal Forwarder traffic was flagged as **Network Service Discovery**

(T1046) demonstrated that legitimate infrastructure activity can resemble adversary behavior. Asset context and analyst validation are critical to avoid unnecessary escalation.

2. **Threat enrichment improves analyst efficiency.**
   Mapping alerts to **MITRE ATT&CK** techniques and associating them with **MITRE D3FEND** defensive actions transformed low-level events into actionable intelligence. This significantly reduced the time required to understand attacker intent and identify relevant response strategies.

3. **Correlation reduces alert fatigue.**
   Aggregating repeated events into higher-level detections (for example, brute-force attempts mapped to **T1110**) reduced noise and enabled focus on sustained attack patterns rather than isolated alerts.

4. **Detection engineering requires continuous tuning.**
   The project highlighted that IDS rules and correlation logic must be refined over time to balance visibility and precision. False positives are inevitable and require iterative tuning based on environment-specific behavior.

5. **Network visibility is critical for IDS effectiveness.**
   Initially, Snort was deployed while all VMs were connected only through a NAT adapter, which prevented visibility into inter-VM traffic and resulted in no alerts. After adding a Host-Only adapter and binding Snort to that interface, alerts were immediately generated. This emphasized the importance of proper sensor placement and network visibility in both virtual and real SOC environments.

6. **Multi-source telemetry increases confidence.**
   Combining Snort network alerts with Sysmon host telemetry provided stronger validation of attack activity than relying on a single data source, reflecting real SOC best practices.

7. **End-to-end implementation builds operational understanding.**
   Designing ingestion, enrichment, correlation, and visualization together reinforced how individual components integrate into a cohesive SOC workflow, mirroring real-world blue team operations.

These lessons demonstrate that effective SOC operations depend not only on tools, but also on context, tuning, and analyst-driven interpretation of threat-informed data.

10. Conclusion

This project successfully designed and implemented a **Threat-Informed SOC Dashboard** using Splunk Enterprise, Snort IDS, and Sysmon, enriched with the MITRE ATT&CK and MITRE D3FEND frameworks.

The lab environment demonstrated that:

- host-based and network-based telemetry can be centrally ingested and correlated,
- raw IDS alerts can be transformed into structured, threat-informed intelligence,
- analysts can rapidly understand attacker intent and defensive options through enriched dashboards.

By simulating realistic adversary activity from a Kali Linux attacker and validating detections across Windows and Ubuntu systems, the project reproduced core SOC workflows including monitoring, triage, correlation, and contextual analysis. The resulting dashboard provided a unified view of adversary behavior and defensive guidance, aligning closely with modern blue team practices.

Overall, this work establishes a practical foundation for threat-informed defense and demonstrates hands-on proficiency in SOC architecture, detection engineering, and security analytics.

11. Future Work

While the current implementation focuses on detection, enrichment, and visualization, several enhancements could further mature the SOC platform:

1. **Automated Response (SOAR) Integration**
   Integrate automated playbooks to trigger actions such as firewall blocking, account lockout, or endpoint isolation for high-confidence detections.
2. **Expanded Data Sources**
   Incorporate firewall logs, VPN logs, Active Directory events, and cloud telemetry to improve detection coverage and context.
3. **Advanced Detection Engineering**
   Develop additional SPL-based behavioral detections using Sysmon data aligned to ATT&CK techniques, beyond signature-based Snort alerts.
4. **Risk-Based Alerting**
   Prioritize detections based on asset criticality, technique severity, and business impact to better support analyst triage.
5. **Broader ATT&CK and D3FEND Coverage**
   Expand mappings to include more tactics and techniques, including persistence, privilege escalation, and lateral movement, as well as preventive D3FEND controls.
6. **SOC Performance Metrics**
   Track metrics such as Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) to evaluate and optimize SOC effectiveness.

7. **Alert Tuning and Whitelisting**
   Implement environment-specific tuning to reduce false positives, especially for trusted infrastructure systems.

These enhancements would transition the lab from a detection-focused prototype into a more autonomous, resilient, and production-aligned SOC environment.