MITRE ATT&CK-Aligned Threat Detection with Splunk

Prepared by: Mythresh Sai Mahadev

Masters in Cybersecurity Operations

Webster University

Project Overview

The "MITRE ATT&CK-Aligned Threat Detection with Splunk" project demonstrates the practical use of Splunk to monitor and detect security threats in real time. By analyzing logs from **auth.log** and **syslog**, the project identifies and maps critical attack techniques such as brute force attempts, account abuse, and file deletions to the MITRE ATT&CK framework. Real-time alerts provide immediate notifications of potential incidents, while an interactive dashboard visualizes key metrics for actionable insights. This project highlights the importance of proactive cybersecurity measures, aligning industry-standard frameworks with practical implementation to effectively combat evolving threats.

Table of Contents

Introduction

The "MITRE ATT&CK-Aligned Threat Detection with Splunk" project implements a security monitoring system using Splunk to detect threats like brute force attacks, account abuse, and file deletions. By analyzing `auth.log` and `syslog` data, the project aligns with the MITRE ATT&CK framework, configures real-time alerts, and visualizes trends through a dashboard, demonstrating effective threat detection and response capabilities.

Objectives

1. Implement a security monitoring system using Splunk to detect and analyze threats in real time.
2. Align detection rules with the MITRE ATT&CK framework to identify key adversary tactics and techniques.
3. Use auth.log and syslog data to simulate and monitor critical attack scenarios.
4. Configure real-time alerts and build dashboards for actionable insights.

5. Deliver actionable insights through Splunk dashboards and real-time alerts.

Step 1: Setting Up Your Splunk Environment

1.1 Install Splunk

1. Download Splunk:

- Visit **Splunk's official website** and create an account if you haven't already.
- Download the latest version of **Splunk Enterprise** for your operating system. Splunk offers a free trial version that should be suitable for this project.

2. Install Splunk:

- **Linux:** Open a terminal and navigate to the folder where you downloaded the .deb package.

```bash
sudo dpkg -i splunk-package-name.deb
```

- **Windows:** Run the installer and follow the on-screen instructions.

3. Start Splunk:

- **Linux:**

```bash
cd /opt/splunk/bin
sudo ./splunk start
```

- **Windows:** Open Splunk from the Start menu.

4. Set Admin Credentials:

- On the first launch, Splunk will ask you to create an admin account. Set a secure username and password, as you'll use these credentials to log in.

5. Log In:

- Open a browser and go to http://localhost:8000.
- Log in with the credentials you set up.

1.2 Verify the Installation

- After logging in, you'll see the Splunk dashboard. This confirms that the installation was successful.

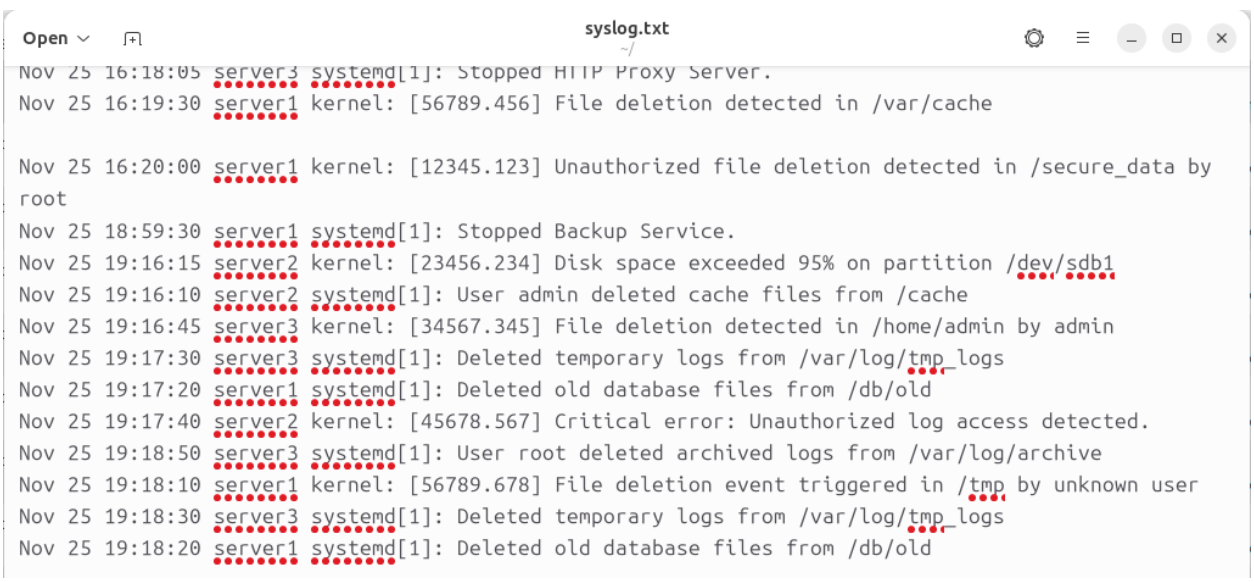
Step 2: Setting Up Data Inputs (Syslog and Auth.log Data)

For basic monitoring, we'll import sample logs to simulate login events and system errors.

2.1 Prepare Sample Data Files

1. Obtain Sample Logs:

- For this project, we'll use two types of logs:
  - **Syslog:** To capture system events.
  - **Auth.log:** To track login attempts.
- You can either download sample syslog and auth.log files or generate them if you have access to logs on a Linux server or you can manually add the sample data as a text file.
- **Syslog file:** Sample log data



- **Auth.log file:** Sample log data.

2. Save and Organize Logs:

- Place these files in a location where Splunk can access them.

2.2 Importing Data into Splunk

1. **Go to Settings > Add Data** in Splunk.
2. Choose Upload and select the syslog and auth.log files you saved.
3. **Set Source Type:**
   - Splunk might automatically detect the source type. If not:
     - Choose ‘syslog’ for the **syslog file**.
     - Choose ‘authlog’ for **auth.log file**.
4. **Set Index:**
   - Create a new index named **mitre_detection** or use the default main index.
5. Click Next and Review your settings, then click **Submit**.

Step 3: Extracting Fields for Action and Severity

Field extraction is necessary to ensure that Splunk can recognize and categorize information like action (for login attempts) and severity (for system errors) within your data. Since Splunk doesn't automatically know these fields from your raw logs, you'll need to create custom field extractions.

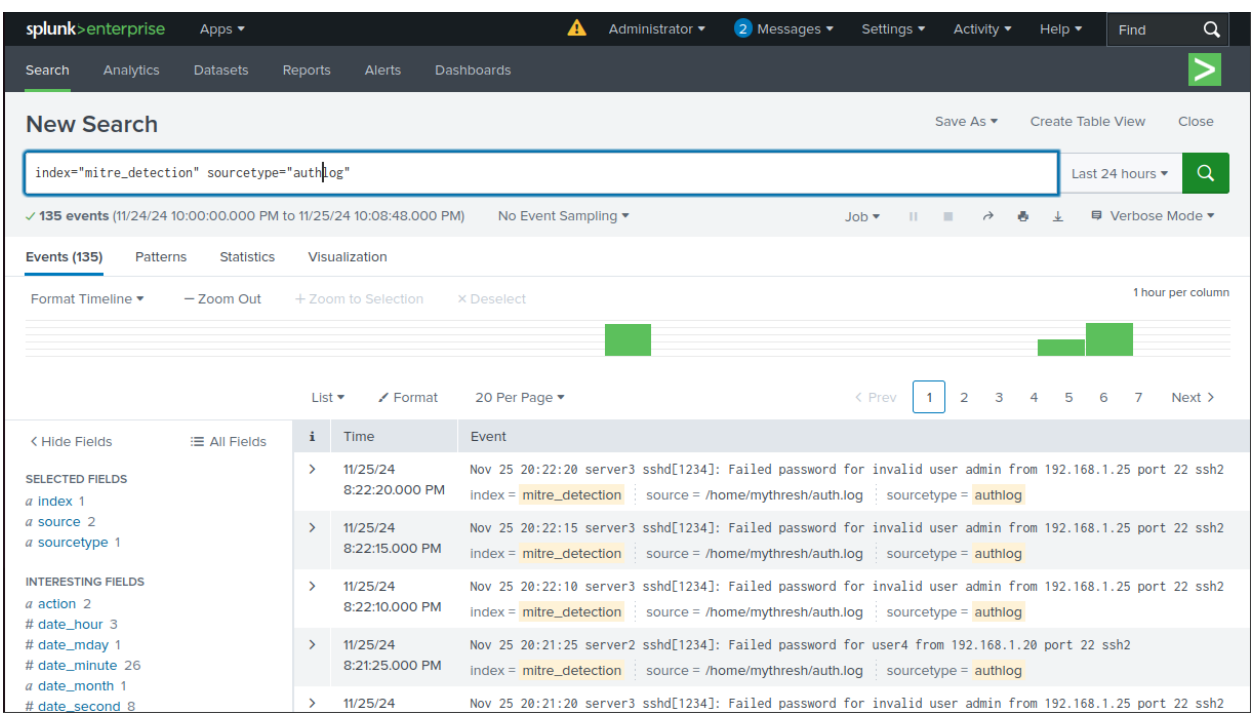
Here's how to extract fields like **action** and **severity**:

3.1 Field Extraction for action (Login Attempts)

1. **Navigate to Search:**
   - Go to Search in Splunk, and search within your authlog data to identify login attempts.
2. **Run a Query to View the Raw Data:**
   - Use the following query to view the raw authlog data:

```spl
index=security_monitoring sourcetype=authlog
```

   - **Search** results using the above query:



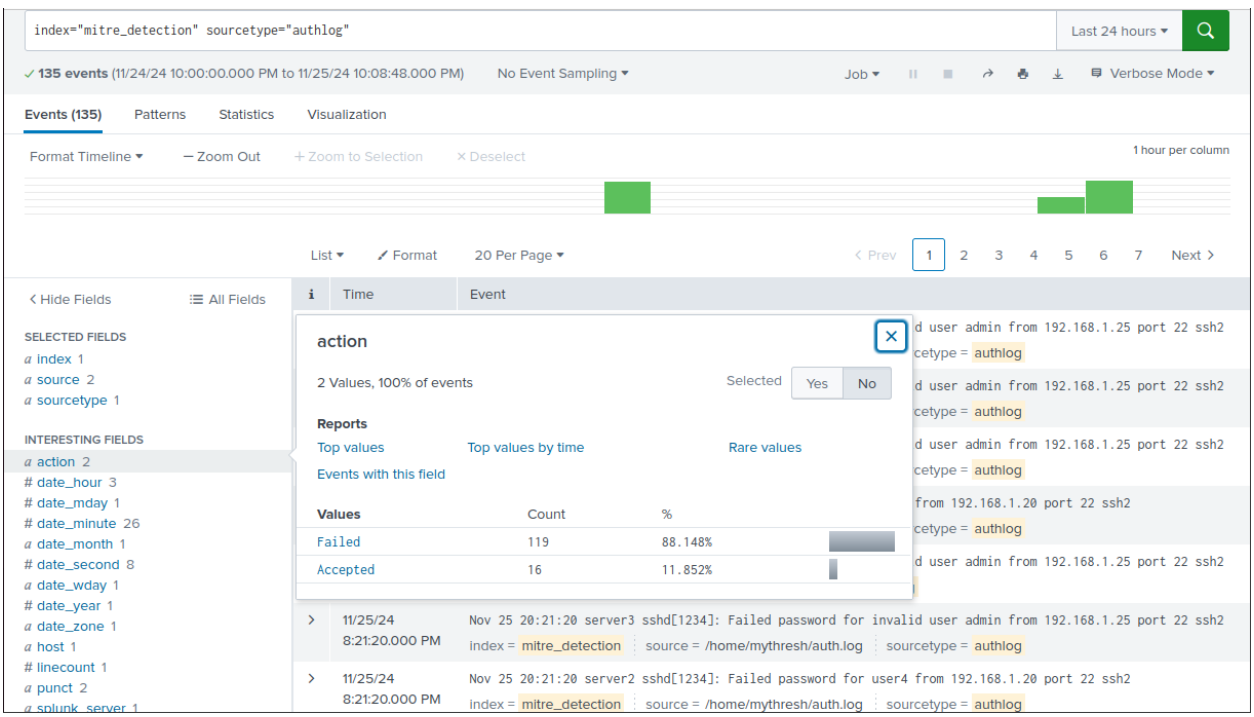
3. Extract the action Field:

- Identify patterns in the raw events, such as keywords like **“failed”** or **“success.”**
- Click on an event, and then on the **Event Actions** menu, select **Extract Fields**.
- Use **Interactive Field Extractor (IFX)**:
  - Highlight the part of the event that specifies the action (e.g., "failed" or "success").
  - Name this field **‘action’**.
  - Splunk will suggest a regular expression; test and refine it to ensure it accurately captures both success and failure.

4. Save the Extraction:

- Test the extraction with various events to confirm accuracy, then save it.

5. After the Extraction:



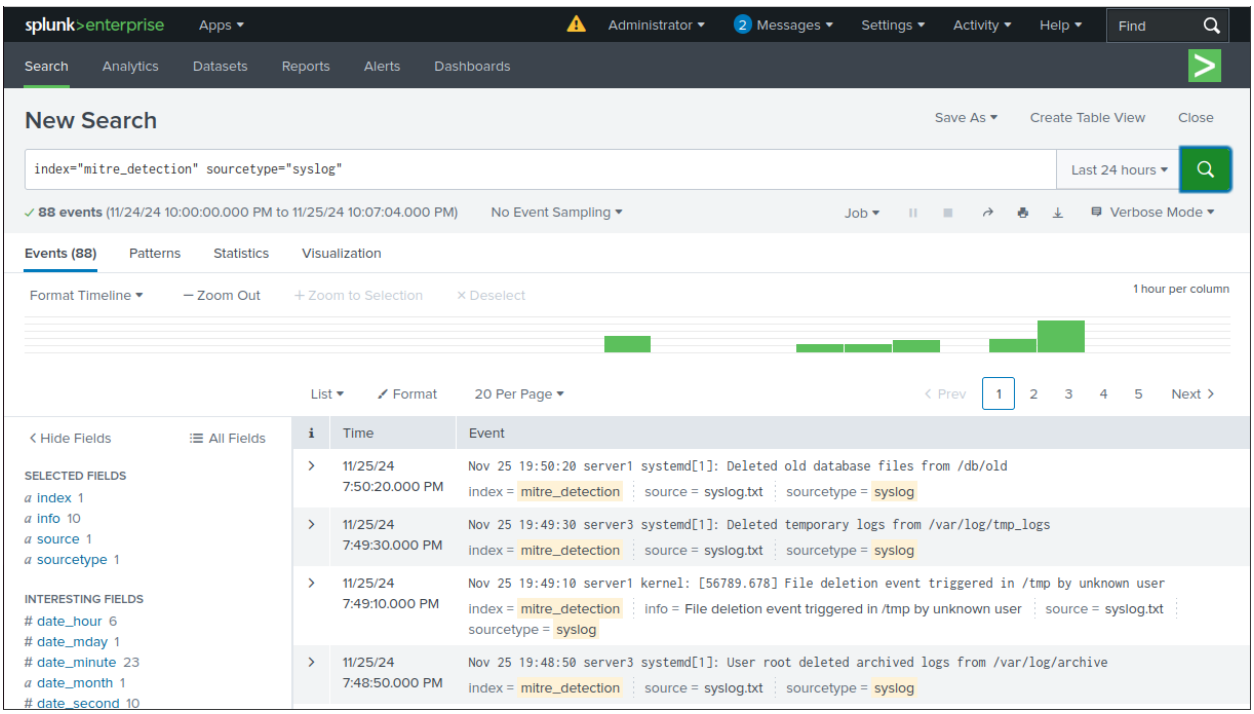
3.2 Field Extraction for severity (System Errors)

1. Search within Syslog Data:

- Use this query to search your syslog data

```spl
index=security_monitoring sourcetype=syslog
```

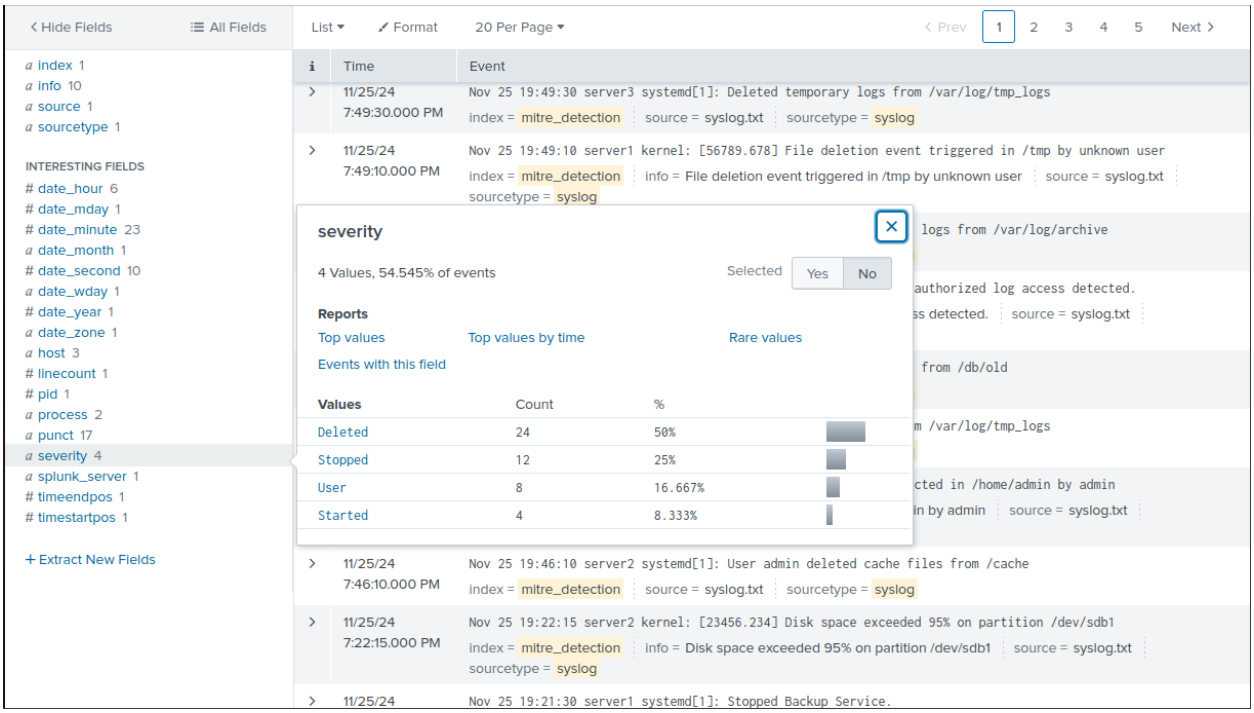
- **Search** results using the above query:



2. Extract the severity Field:

- Look for keywords that indicate severity levels, such as **"error," "warning," or "info."**
- Use the same **Extract Fields** tool to isolate the severity information.
- Name this field **severity** and ensure the regular expression captures all relevant severity levels.

3. **After the Extraction:**



4. **Test and Save**:

- Test the extraction with various events to confirm accuracy, then save it.

Step 4: Map a Threat Group to MITRE ATT&CK

This step involves selecting a threat group or campaign and mapping their TTPs (Tactics, Techniques, and Procedures) using the MITRE ATT&CK framework.

4.1. Choose a Threat Group

- Select a known adversary that aligns with your logs and the MITRE ATT&CK framework.
- **Example Threat Group**: APT28 (Fancy Bear)
    - **Reason**: They are known for brute-force attacks and credential theft, which relate to the auth.log and syslog events we set up.

4.2. Research the Threat Group's TTPs

Use reliable sources like the MITRE ATT&CK knowledge base:

- **APT28 Overview**: MITRE ATT&CK - APT28
- Review the techniques they commonly use, such as:
    - **Initial Access**: T1078 (Valid Accounts)
    - **Execution**: T1059 (Command and Scripting Interpreter)
    - **Persistence**: T1136 (Create Account)
    - **Privilege Escalation**: T1110 (Brute Force)
    - **Defense Evasion**: T1070 (Indicator Removal on Host)

4.3. Use the MITRE ATT&CK Navigator

The Navigator tool helps visualize the TTPs.

1. **Access the Navigator**:
    - Go to MITRE ATT&CK Navigator.
2. **Create a Heatmap**:
    - Highlight techniques associated with APT28.
    - Save the heatmap as a JSON or image file for documentation.

Step 5: Define Detection Rules

This involves writing Splunk queries to detect suspicious activities and setting up alerts.

5.1. Start with TTPs Mapping

For each technique you mapped, identify the corresponding log data and write detection rules.

- Summarize mapped techniques in a table:

Technique ID	Technique Name	Tactic	Description
T1110	Brute Force	Credential Access	Detects repeated failed login attempts.
T1078	Valid Accounts	Persistence	Detects misuse of valid credentials.
T1070	Indicator Removal on Host	Defense Evasion	Detects deletion of logs to evade detection

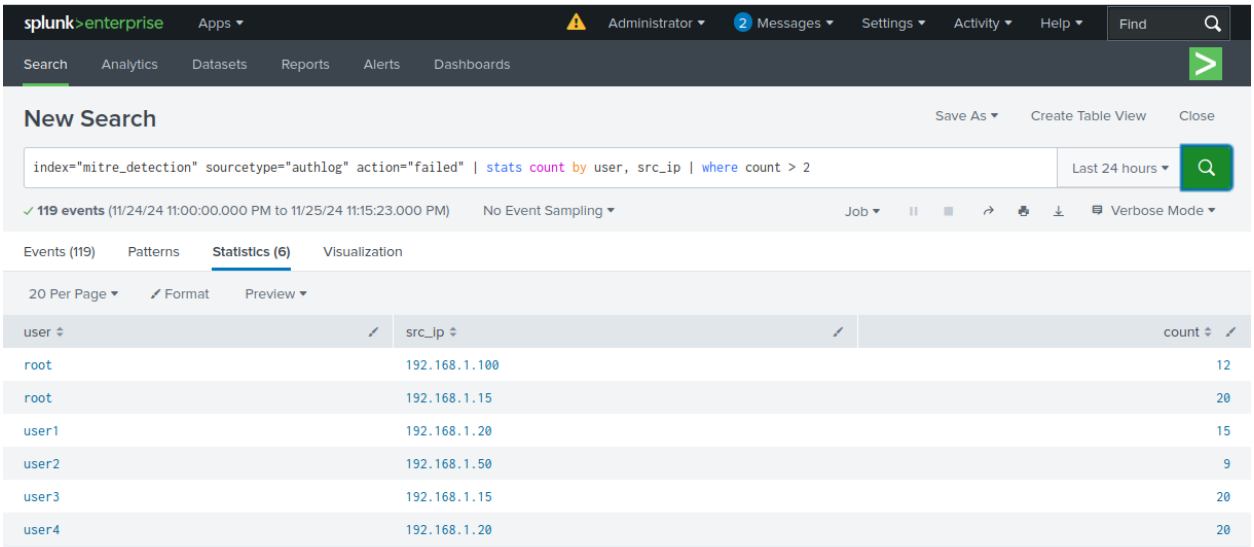
5.2. Write Splunk Search Queries

Here are queries of detection rules for logs:

- **Query 1: Detect Failed Login Attempts (Brute Force - T1110):**

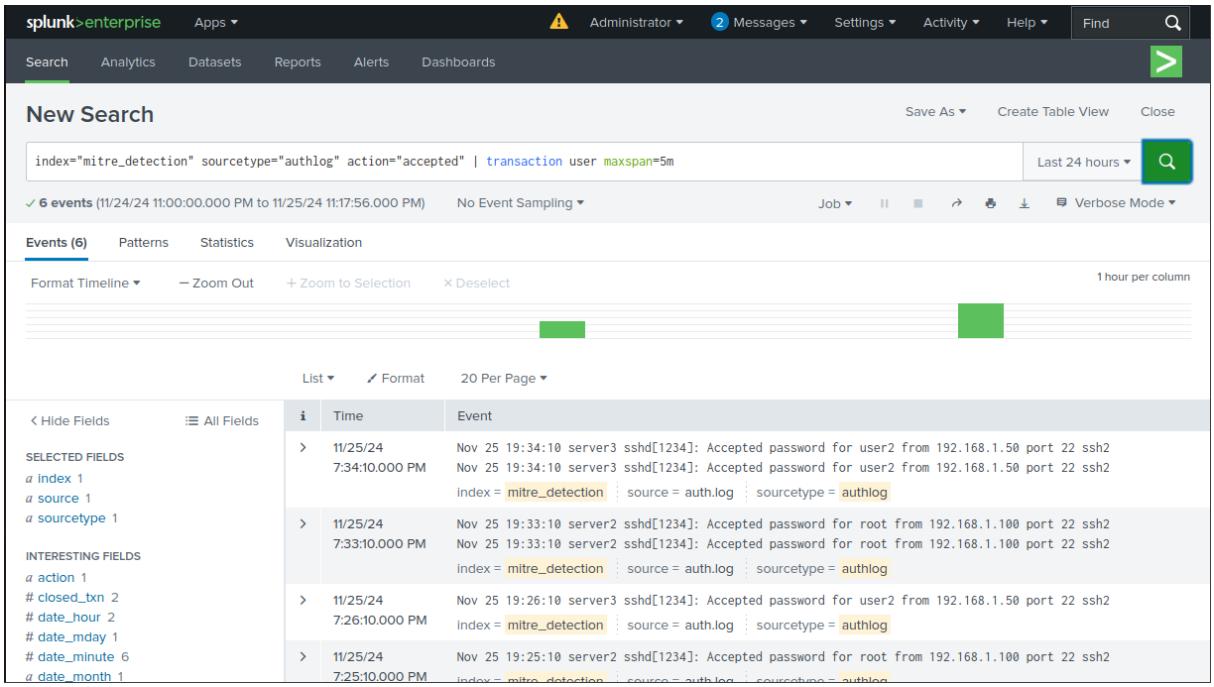
```spl
index=mitre_detection sourcetype=authlog action="failed"
| stats count by user, src_ip
| where count > 2
```

  - **Explanation**: This query identifies users or IP addresses with more than 2 failed login attempts, indicating a potential brute-force attack.



- **Query 2: Detect Successful Logins After Multiple Failures (Valid Accounts - T1078):**
  - **Explanation**: This query detects successful logins for a user after multiple failures within 5 minutes.

```spl
index=mitre_detection sourcetype=authlog action="accepted"
| transaction user maxspan=5m
```

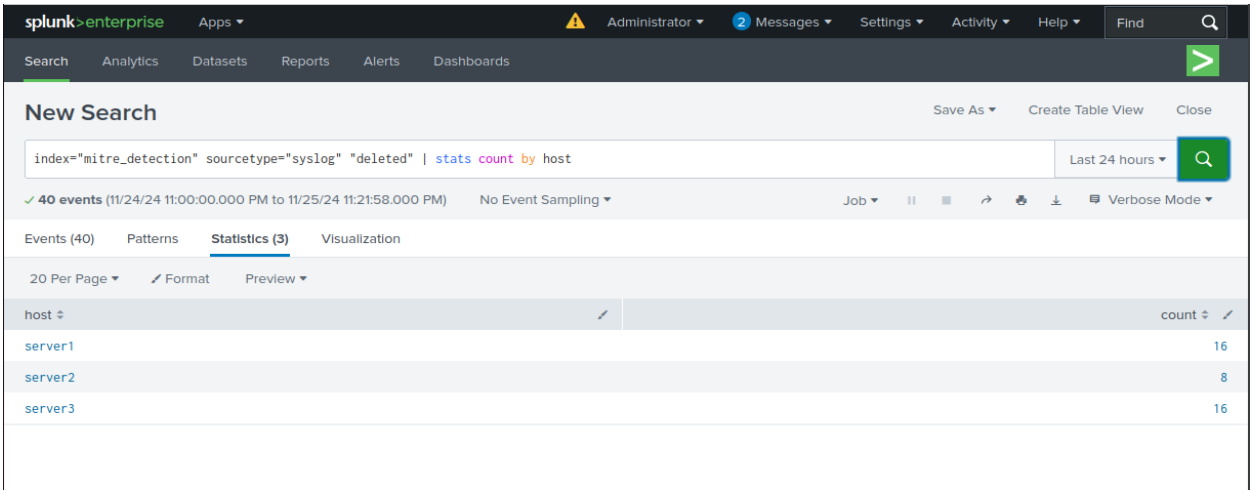


- **Query 3: Detect File Deletions (Indicator Removal - T1070):**

spl Copy code

```spl
index=mitre_detection sourcetype=syslog "deleted"
| stats count by host
```

- **Explanation**: This query flags events where files were deleted, which could indicate attempts to hide malicious activity.



Step 6: Setting Up Alerts

Alerts will notify you of significant events, such as multiple failed login attempts and system errors. We'll configure basic alerts for common security monitoring scenarios.

6.1 Create a New Index for Alerts

1. **Navigate to Settings > Indexes**:
    - Go to the main Splunk dashboard, then **Settings** > **Indexes**.
2. **Create a New Index**:
    - Click on **New Index** and name it something like alert_logs.
    - Set any necessary retention policies (optional) based on how long you want to keep the alert data.
3. **Save the Index**:
    - Click **Save** to finalize your new index.

6.2 Configure an Alert for Detect Brute Force Attempts – T1110

1. **Create a Search for Failed Logins**:

```spl
index=mitre_detection sourcetype=authlog action="failed"
| stats count by user, src_ip
| where count > 2
```

- Go to **Search & Reporting** and use the following query to find failed login attempts.
- This query isolates all failed login attempts in your authlog data.

2. **Set Conditions for the Alert**:
    - Click on **Save As** in the upper right corner and choose **Alert**.
    - **Alert Title**: Name it "**Detect Brute Force Attempts – T1110**"
    - **Alert Type**: Select **Scheduled** if you want it to check periodically, or **Real-time** for instant detection.
    - **Trigger Conditions**:
        - Set the condition to trigger when there are, for example, **Per-Result**

3. **Configure Alert Actions**:
   - In the **Alert Actions** section, choose **Log Event**.
   - **Event box: "`Brute Force Detected`".**
   - **Destination Index**: Specify the `alert_logs` index. This index is where all triggered alert events will be stored.

4. **Save the Alert**.

6.3 Configure an Alert for Detect File Deletion – T1070

1. **Create a Search for System Errors**:

   - Use this search to capture error-level system logs:

```spl
index=mitre_detection sourcetype=syslog "deleted"
| stats count by host
```

   - This will isolate any events labeled as errors in the syslog data.

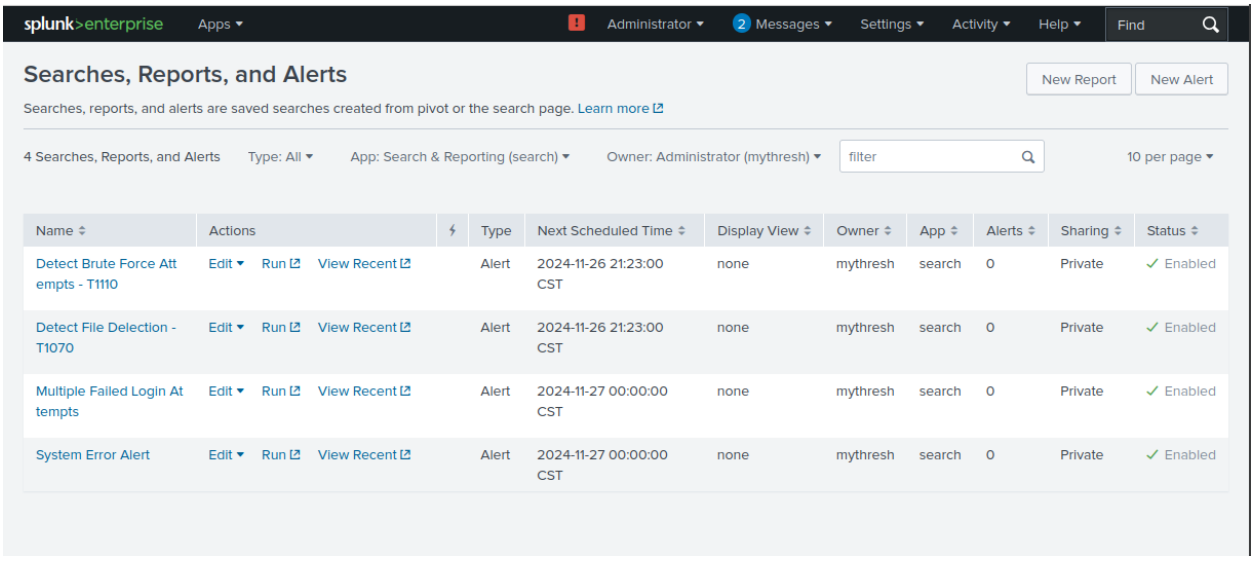
2. **Set Conditions for the Alert**:

   - Click on **Save As** in the upper right corner and choose **Alert**.
   - **Alert Title**: Name it "**Detect File Deletion – T1070**"
   - **Alert Type**: Select **Scheduled** if you want it to check periodically, or **Real-time** for instant detection.
   - **Trigger Conditions**:
     - Set the condition to trigger when there are, for example, **3 or more failed login attempts within 5 minutes**.

3. **Configure Alert Actions**:
   - In the **Alert Actions** section, choose **Log Event**.
   - **Event box: "`Alert! File Deletion Detected`"**
   - **Destination Index**: Specify the `alert_logs` index. This index is where all triggered alert events will be stored.

4. **Save the Alert**.

6.4 Saved Alerts:

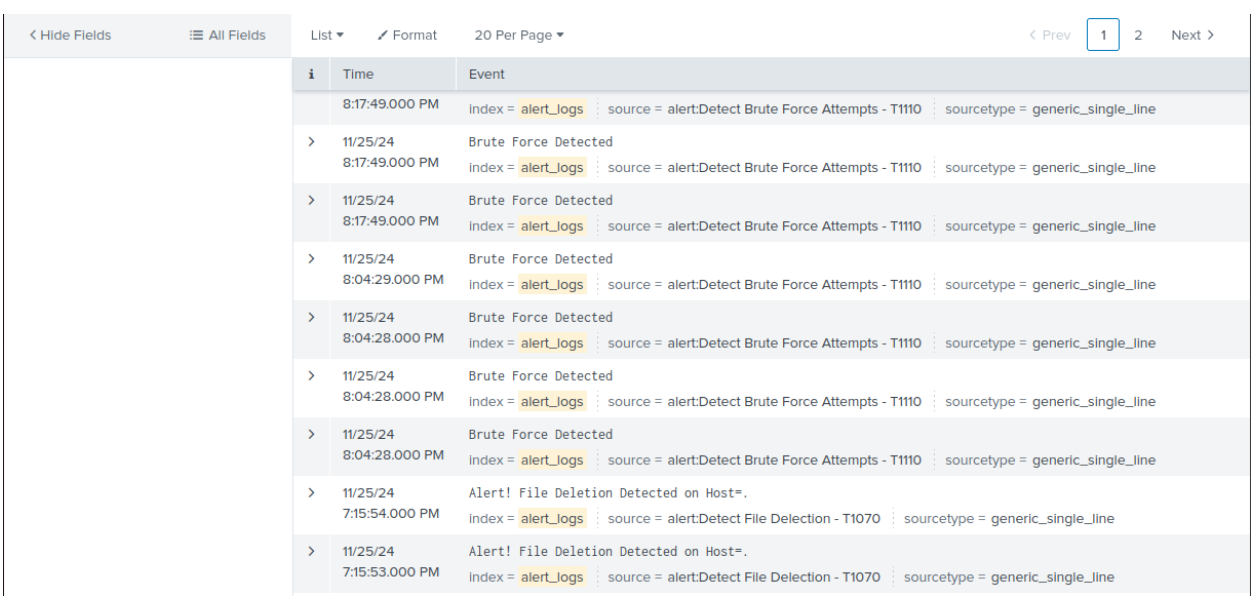


6.5 Reviewing Alerts

Once saved, if the condition triggers as per the alert configuration then you can view your alert events in the **`alert_logs`** index using:

```spl
index=alert_logs
```

Here are the **Search** results using the above query:



Step 7: Dashboard Creation

To monitor security events effectively, we will create a dashboard that visualizes critical metrics, such as failed login attempts, successful login attempts, and system errors. The dashboard will provide immediate insights into potential security threats.

7.1 Create a New Dashboard

1. **Navigate to Dashboards**:
    - In Splunk's main menu, go to **Dashboards** and click **Create New Dashboard**.
2. **Name Your Dashboard**:
    - Enter a meaningful name, such as "**Threat Detection Dashboard**".
    - Choose a suitable app, such as **Search & Reporting**, and set the permissions to **Shared in App**.
3. **Select Dashboard Studio**:
    - Choose **Classic Dashboard Studio** for flexibility in design and layout.

7.2 Add Panels to Display Key Metrics

We will add panels to visualize:

- **Brute Force Detection**
- **Valid Account Abuse**
- **File Deletion Detection**
- **Alert Logs Overview**

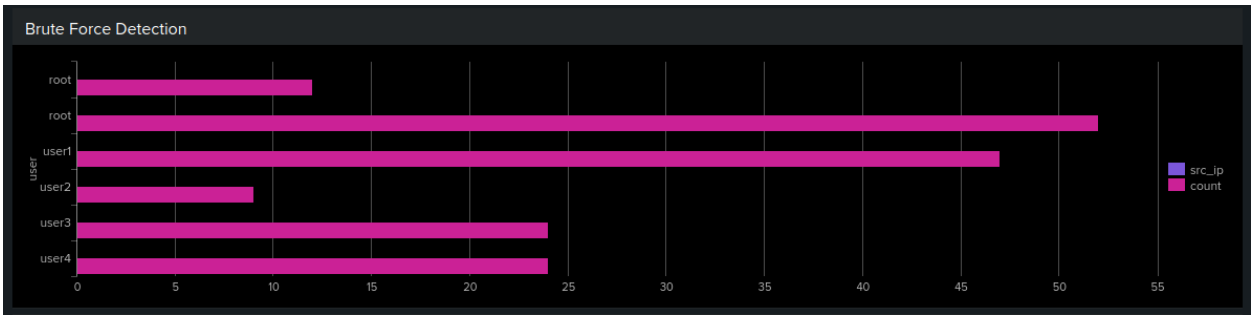
Panel 1: Brute Force Detection

1. **Add New Panel**:
    - Click **Add New Panel** in the dashboard editor.
2. **Search Query**

spl Copy code

```spl
index=mitre_detection sourcetype=authlog action="failed"
| stats count by user, src_ip
| where count > 2
```

3. **Visualization**:
    - Select a **Bar Chart** to show the count of failed login attempts per user or IP.
4. **Title**:
    - Set the title to "**Brute Force Detection**".

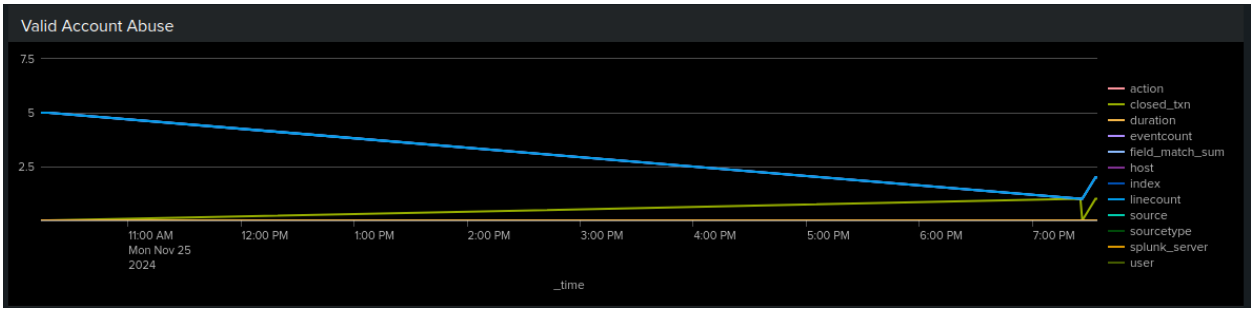


Panel 2: Valid Account Abuse

1. **Add New Panel**:
    - Click **Add New Panel** in the dashboard editor.
2. **Search Query**

```spl
index=mitre_detection sourcetype=authlog action="accepted"
| transaction user maxspan=5m
```

3. **Visualization**:
    - Select an **Area Chart** to show the count of failed login attempts per user or IP.
4. **Title**:
    - Set the title to "**Valid Account Abuse**".



Panel 3: File Delete Detection

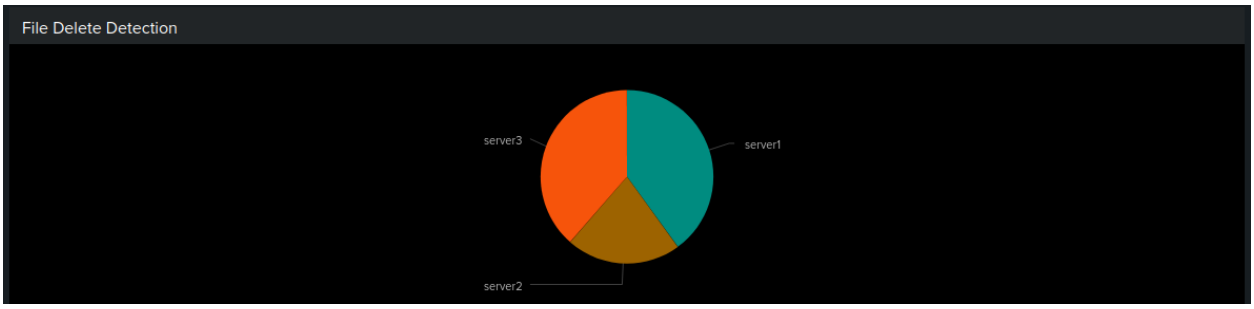
1. **Add New Panel**:

- Click **Add New Panel** in the dashboard editor.

2. **Search Query**

```spl
index=mitre_detection sourcetype=syslog "deleted"
| stats count by host
```

3. **Visualization**:
   - Select a **Pie Chart** to show the count of failed login attempts per user or IP.
4. **Title**:



   - Set the title to "**File Delete Detection**".

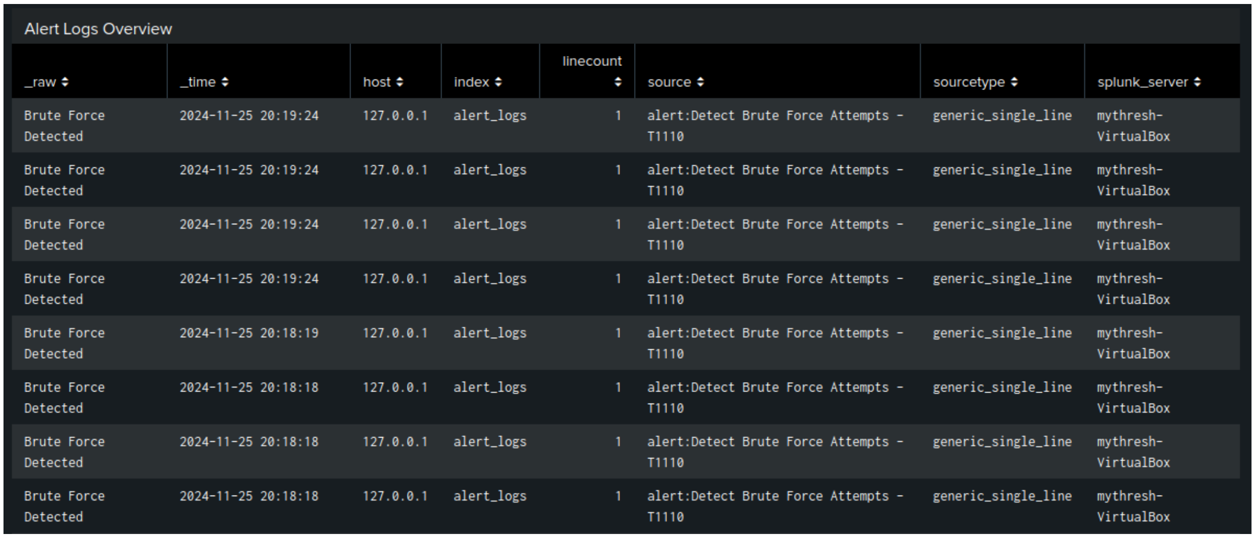
Panel 4: Alert Logs Overview

1. **Add New Panel**:
   - Click **Add New Panel** in the dashboard editor.
2. **Search Query**

```spl
index=alert_logs
```

3. **Visualization**:
   - Select a **Bar Chart** to show the count of failed login attempts per user or IP.

- **Title**: Set the title to “**Alert Logs Overview**”.



Alert Logs Overview

_raw ⇕	_time ⇕	host ⇕	index ⇕	linecount ⇕	source ⇕	sourcetype ⇕	splunk_server ⇕
Brute Force Detected	2024-11-25 20:19:24	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:19:24	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:19:24	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:19:24	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:18:19	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:18:18	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:18:18	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox
Brute Force Detected	2024-11-25 20:18:18	127.0.0.1	alert_logs	1	alert:Detect Brute Force Attempts - T1110	generic_single_line	mythresh-VirtualBox

7.3 Save and Review the Dashboard

1. After adding all panels, arrange them for readability.
2. Save the dashboard and test the panels to ensure data is displayed as expected.

Conclusion

The "MITRE ATT&CK-Aligned Threat Detection with Splunk" project successfully demonstrates the implementation of a robust security monitoring system using Splunk. By analyzing logs from `auth.log` and `syslog`, critical threats such as brute force attacks, account abuse, and file deletions were detected and mapped to the MITRE ATT&CK framework. Real-time alerts provided immediate notifications of incidents, while a comprehensive dashboard visualized security trends and insights effectively.

This project highlights the importance of proactive threat detection and monitoring in cybersecurity. It showcases advanced log analysis, real-time alerting, and visualization capabilities, making it a scalable and impactful solution for real-world security challenges.