**Understanding log messages**

1. Syslog

Jun 11 2024 11:52:42 GPN-0003\_TRC-CR\_X8-01 %%01OSPF/4/CONFLICT\_ROUTERID\_INTF(l):CID=0x8082047e;OSPF router ID conflict is detected on the interface.(ProcessId=801, RouterId=10.0.0.10, AreaId=0.0.0.0, InterfaceName=GigabitEthernet6/0/11.801, IpAddr=10.0.0.22, PacketSrcIp=10.0.0.10)

Timestamp: Jun 11 2024 11:52:42

Hostname: GPN-0003\_TRC-CR\_X8-01; GPN-vendor, 0003- site ID. at TRC location, location code- TRC (CP for CapeTown), CR-core (core level router, others are: access ACC, aggregator AG, metrocore, routereflector RR, X8-huawei router X8 series (model number for router). M14: NetEngine 8000 Huawei. S97: Switch 9700 series from Huawei.

Cisco/Huawei identifier: %%

Primary router: 01 (M14-01)

Secondary router: 02 (M14-02) it can be backup for primary router or also have some exclusive responsibilities like load balancing to work in tandem with primary router.

Module Name: OSPF

Severity: 4

logType: CONFLICT\_ROUTERID\_INTF

description: (l):CID=0x8082047e;OSPF router ID conflict is detected on the interface.(ProcessId=801, RouterId=10.0.0.10, AreaId=0.0.0.0, InterfaceName=GigabitEthernet6/0/11.801, IpAddr=10.0.0.22, PacketSrcIp=10.0.0.10)

CID: circuit ID. Identification for any connection/circuit

10 ports on a router, each port is an interface. Similar to laptop.

Protocols: SGP, OSPF, etc. these contain submodules as well.

OSPF specific metadata: ProcessId=801, RouterId=10.0.0.10, AreaId=0.0.0.0, InterfaceName=GigabitEthernet6/0/11.801, IpAddr=10.0.0.22, PacketSrcIp=10.0.0.10

Socket: connection bw client and server

MD5\_Authentication\_Fail: MD5 level of authentication, which is sent along with TCP protocol

Jun 11 2024 11:52:46 GPN-0001\_TLK-CR\_M14-01 %%01SNMP/4/SNMP\_IPUNLOCK(s):CID=0x80d50422;The source IP was unlocked.(SourceIP=154.73.157.104, VPN=SKYWIRE.VPN)

Jun 11 2024 11:26:54 GBN-0001\_TLK-CR\_M14-02 %%01TCP/4/SOCKET\_TCP\_PACKET\_MD5\_AUTHEN\_FAIL(l):CID=0x80650553;TCP MD5 authentication failed. (tcpConnLocalAddress=172.31.0.103, tcpConnLocalPort=54291, tcpConnRemAddress=172.31.0.9, tcpConnRemPort=179, hwTCPProtocol=BGP, hwTCPVrfName=\_public\_)

4897: Jun 11 2024 12:10:34 GBN-0002\_GST-CR\_M14-01 %%01CLI/5/LOGINFAILED(s):CID=0x80ca2713;The user failed to log in to VTY1. (UserType=Telnet, UserName=supportadmin, RemoteIp=182.186.34.0, VpnName=GPG.INT, Reason=local user does not exist, LocalIp=66.8.6.65)

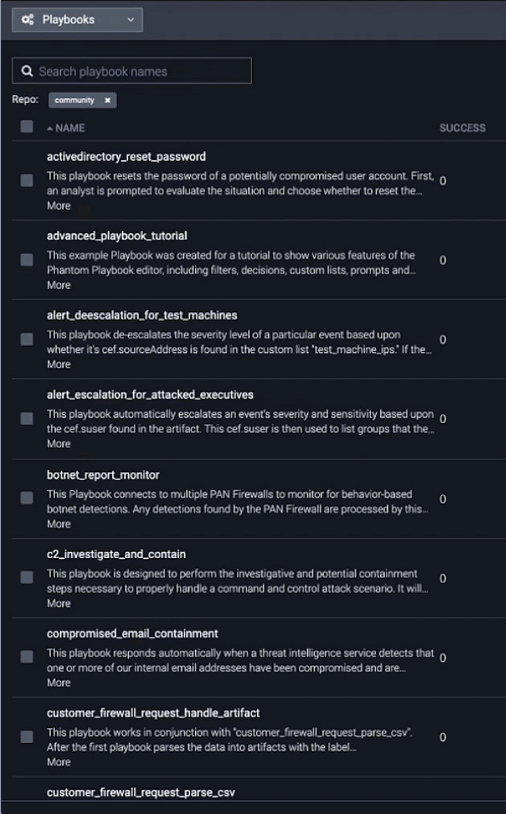
29: Jun 11 2024 11:53:06 GPN-0277\_DAV-CLI\_S57-01 %%01FTPS/5/REQUEST(l)[66038]:The user had a request. (UserName="", IpAddress=172.31.251.24, VpnInstanceName="", Request=QUIT

)

1. TACACS

SOAR Playbook: the goal of your playbook should not just be automation for the sake of automation. It should be to enhance your security team’s efficiency

1. Start by identifying repetitive tasks and processes that can be automated
2. define clear metrics to measure the effectiveness of your playbook, such as response times and false positive rates.
3. Example playbooks from Splunk SOAR:



**ML for pattern recognition:**

1. Manual labeling for known patterns
   1. **Define the Problem**
      1. Identify the types of correlations you want to detect, such as:
      2. Failures in routers causing failures in connected routers.
      3. Correlations between high traffic and router errors.
   2. Normalise the data and extract relevant features from each source
      1. Extract features from each dataset that are relevant for correlation into one unified dataset. For example, in syslog data, you might extract error codes and timestamps. In traffic data, you might extract traffic volume and types.
      2. If datasets cannot be merged directly, you can perform time-based correlation. Match events based on their timestamps, and analyze how events from different datasets align over time.
      3. Cross-correlation: Create a system to cross-reference events between different datasets based on identified features. This might involve creating rules (manual) or using similarity measures (algorithms).
2. There are several algorithms and techniques that reduce or eliminate the need for manual data labeling:

**2.1. Unsupervised Learning**

**Clustering**: As mentioned earlier, clustering algorithms do not require labeled data. They group similar data points together, helping to discover patterns and anomalies.

**Dimensionality Reduction**: Algorithms like PCA or t-SNE can help in identifying important features and relationships in your data without needing labels.

**2.2. Self-Supervised Learning**

**Self-Supervised Learning**: This technique involves creating supervisory signals from the data itself, reducing the need for manual labeling. Models learn to predict parts of the data from other parts.

Example: Predicting missing parts of a log entry based on the rest of the entry can help the model learn patterns without explicit labels.

**2.3. Semi-Supervised Learning**

**Semi-Supervised Learning**: Uses a small amount of labeled data along with a large amount of unlabeled data. The labeled data helps guide the model, while the unlabeled data provides additional information for better learning.

Example: You might label a small sample of critical errors, and the model learns to generalize from this small labeled set along with a large amount of unlabeled log data.

**2.4. Reinforcement Learning**

**Reinforcement Learning**: This approach learns by interacting with the environment and receiving feedback. It can be used for tasks like optimizing network performance or automating response strategies without needing extensive labeled datasets.

Example: An RL agent could learn to identify and respond to network issues based on feedback from system performance rather than predefined labels.

**Cross-Dataset Correlation:**

1. **Time-Based Correlation**

Even if datasets can’t be merged directly, you can correlate events based on time windows or intervals.

Use similarity measures to compare patterns between datasets.

If the correlation patterns are complex, train models to learn correlations between different features across datasets.

Automate the process of correlation and pattern recognition and visualize the results to identify actionable insights.

1. **Non-Time-Based Correlation:**

If you don’t want to rely on time-based correlation for events that may be related but occur far apart, you can explore other methods to identify correlations between disparate datasets. Here are some alternative approaches:

1. **Feature-Based Correlation**
   1. **Create a Common Feature Space**

Identify key features across datasets that can be used to find relationships. You can then use these features to establish correlations.

Example: If you have network traffic data and syslog data, both might contain information about error codes or types of anomalies. Create a common feature space where you compare these features.

* 1. **Use Similarity Measures**

Apply similarity measures to compare features across datasets.

1. **Cross-Feature Correlation**
   1. **Build Correlation Models**

Create models to predict one feature based on features from other datasets. For example, predict traffic anomalies based on syslog data.

**2.2. Use Embeddings**

Use embeddings to represent data from different datasets in a common vector space, enabling comparison and correlation.

1. **Graph-Based Approaches**
   1. **Build a Graph of Events**

Represent events as nodes in a graph and use graph algorithms to find relationships. For example, nodes could represent different types of events, and edges could represent correlations.

* 1. **Community Detection**

Use community detection algorithms to find clusters of related events across datasets.

1. **Anomaly Detection and Pattern Recognition**

**4.1. Unsupervised Anomaly Detection**

Apply anomaly detection techniques to find unusual patterns in each dataset, then correlate anomalies across datasets.

**4.2. Pattern Recognition**

Train pattern recognition models to find complex patterns in data across datasets.

1. **Automation and Reporting**

**5.1. Automate Correlation Process**

Implement a pipeline to automatically handle new data, perform preprocessing, and apply correlation models.

5.2. **Visualization and Insights**

Visualize correlations and patterns to derive actionable insights.

**Summary**

To identify correlations and patterns without merging datasets directly or relying on time windows:

1. **Feature-Based Correlation**: Create a common feature space and use similarity measures.
2. **Cross-Feature Correlation**: Build models to predict features across datasets or use embeddings.
3. **Graph-Based Approaches**: Construct a graph of events and apply community detection.
4. **Anomaly Detection and Pattern Recognition**: Use unsupervised learning to detect anomalies and recognize patterns.
5. **Automation and Reporting**: Implement a pipeline for automated processing and visualization.

These methods enable you to find correlations and patterns even when datasets are stored in separate directories and can't be combined directly.

**Event Correlation across multiple datasets**

To perform \*\*event correlation\*\* across three contextually linked datasets, you need an algorithm that can identify and analyze relationships between events, even when they are spread across different sources. The choice of algorithm depends on the type of data, the relationships between them, and the structure of the events. Here are a few techniques that could be useful for this kind of task:

### 1. \*\*Graph-Based Algorithms (Graph Mining)\*\*

- \*\*Objective\*\*: Represent the datasets as nodes (events) and edges (relationships between events) to identify correlations and patterns across them.

- \*\*Technique\*\*:

- \*\*Graph Databases (e.g., Neo4j)\*\*: You can model your data as a graph where events are nodes, and contextual relationships between the datasets are edges. You can then use algorithms like \*\*community detection\*\*, \*\*shortest path\*\*, or \*\*graph traversal\*\* to find correlations between events.

- \*\*PageRank or Personalized PageRank\*\*: To prioritize certain types of events or connections that are contextually important.

\*\*When to use\*\*: When the datasets are complex, and the relationships between events can be naturally modeled as a graph (e.g., networks, systems logs).

### 2. \*\*Causal Inference Models\*\*

- \*\*Objective\*\*: Determine the cause-effect relationship between events across datasets.

- \*\*Techniques\*\*:

- \*\*Granger Causality\*\*: A statistical hypothesis test for determining whether one time series can predict another, making it useful for event-driven data where one event may lead to another across datasets.

- \*\*Bayesian Networks\*\*: These can model the probabilistic relationships between events and help identify causal links. Useful when there's uncertainty about the connections between datasets.

\*\*When to use\*\*: When you are interested in discovering potential causal relationships between events in the different datasets.

### 3. \*\*Cross-Dataset Association Rule Learning\*\*

- \*\*Objective\*\*: Find patterns and correlations (association rules) between events spread across different datasets.

- \*\*Techniques\*\*:

- \*\*Apriori or FP-Growth Algorithms\*\*: Used to find frequent itemsets and association rules. You can treat events as items and datasets as "baskets" to identify common events that occur together.

- \*\*Sequential Pattern Mining\*\*: This technique extends association rules to find correlations between events that occur in a particular order, which could help you identify sequences of related events across datasets.

\*\*When to use\*\*: When you're interested in finding frequent co-occurring events or patterns across datasets.

### 4. \*\*Temporal Correlation (Time Series Analysis)\*\*

- \*\*Objective\*\*: Correlate events based on their time of occurrence.

- \*\*Techniques\*\*:

- \*\*Dynamic Time Warping (DTW)\*\*: A technique that aligns time series data from different sources, even if they are out of sync, and finds similarities or correlations.

- \*\*Cross-Correlation\*\*: Measures the similarity between two time series, which can be extended to multiple datasets to find temporal correlations between events across datasets.

\*\*When to use\*\*: When events have timestamps and you want to correlate events based on when they occurred.

### 5. \*\*Clustering with Contextual Data Integration\*\*

- \*\*Objective\*\*: Group events from different datasets that are contextually similar.

- \*\*Techniques\*\*:

- \*\*Hierarchical Clustering\*\* or \*\*K-Means\*\*: You can merge datasets based on contextual features, and then use clustering algorithms to group events that are similar across the three datasets.

- \*\*Affinity Propagation\*\*: A clustering method that can be used when relationships between events are more complex and context-based.

\*\*When to use\*\*: When you want to group events based on similarities and relationships, particularly when events are linked by context rather than direct features.

### 6. \*\*Latent Semantic Analysis (LSA) or Latent Dirichlet Allocation (LDA)\*\*

- \*\*Objective\*\*: Find underlying patterns or topics in events across multiple datasets by representing the contextual linkages as latent topics.

- \*\*Techniques\*\*:

- \*\*LSA\*\*: Use singular value decomposition (SVD) to identify the latent structure in the event data. This is particularly useful if the events are textual or have categorical context.

- \*\*LDA\*\*: A probabilistic model that finds underlying topics (patterns) in event data, especially useful if the context is language-based or unstructured.

\*\*When to use\*\*: If the datasets are event logs, documents, or records with a lot of unstructured data, and you want to find patterns by extracting latent themes.

### 7. \*\*Matrix Factorization Techniques\*\*

- \*\*Objective\*\*: Uncover latent patterns between events across datasets by factorizing event-context matrices.

- \*\*Technique\*\*:

- \*\*Non-negative Matrix Factorization (NMF)\*\*: Useful for finding latent structures by factorizing data into parts-based representations. This can help identify hidden event correlations when data is spread across different sources but has common contexts.

\*\*When to use\*\*: When you have a matrix of events versus contextual features and you want to find hidden patterns across the datasets.

---

### Recommendation:

If your datasets are contextually linked but not directly, \*\*graph-based approaches\*\* or \*\*causal inference models\*\* are likely the best options. These techniques can handle complex, multi-relational data and find correlations across datasets based on contextual or indirect relationships.

If the events are time-sensitive, \*\*temporal correlation methods\*\* like \*\*Dynamic Time Warping (DTW)\*\* or \*\*cross-correlation\*\* would be a good fit.

For large-scale or high-dimensional data, \*\*latent factor models\*\* like \*\*LSA\*\* or \*\*matrix factorization\*\* can be effective in uncovering hidden patterns and relationships.